НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Факультет Інформатики та обчислювальної техніки
Кафедра Автоматики та управління в технічних системах

«До захисту допущено»
Завідувач кафедрою
_____ Ролік О. І.
(підпис) (ініціали, прізвище)
«__»_____ 2019 р.

Магістерська дисертація

зі спеціальності 126 Інформаційні системи та технології

на тему: «Система управління IoT мережею»

Виконав: студент 6-го курсу, групи ІА-82мп
(шифр групи)

Шлапак Сергій Сергійович
(прізвище, ім'я, по батькові) (підпис)

Науковий керівник д.т.н, професор каф. АУТС, Корнієнко Б. Я.
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Рецензент _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2019 року

Національний технічний університет України
«Київський політехнічний інститут
імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки
(повна назва)

Кафедра автоматики та управління в технічних системах
(повна назва)

Рівень вищої освіти – другий (магістерський) за освітньо-професійною програмою

Спеціальність 126 Інформаційні технології та системи
(код і назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Ролік О. І.
(підпис) (ініціали, прізвище)
«__» _____ 2018 р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Шлапаку Сергію Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема дисертації «Система управління IoT мережею»

науковий керівник дисертації Корнієнко Богдан Ярославович, к.т.н.,
(прізвище, ім'я, по-батькові, науковий ступінь, вчене звання)
доцент кафедри АУТС

затверджені наказом по університету від «___» _____ 2019 р. №___

2. Строк подання студентом дисертації _____

3. Об'єкт дослідження система управління IoT мережею

4. Предмет дослідження методи захисту мережі та приватної інформації користувача

5. Перелік завдання, які потрібно розробити огляд існуючих рішень, порівняння стандартів захисту, модель системи, графічний матеріал

6. Орієнтовний перелік ілюстративного (графічного) матеріалу: схема структурна, ER-діаграми, діаграма станів, діаграма прецедентів, приклади роботи алгоритмів та системи

7. Орієнтовний перелік публікацій: «Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення»,

8. Консультанти розділів дисертації

9. Дата видачі завдання – 02.09.2019

Календарний план

№ з/п	Назва етапів виконання дипломного проекту	Строк виконання етапів проекту	Примітка
1.	Огляд існуючих рішень	2.09.2019 р.	
2.	Опис та математична модель алгоритмів	14.09.2019 р.	
3.	Модель для порівняння алгоритмів	28.09.2019 р.	
4.	Аналіз результатів та вибір архітектури	1.10.2019 р.	
5.	Розроблення моделі системи	15.10.2019 р.	
6.	Аналіз отриманих результатів	31.10.2019 р.	
7.	Розроблення схеми структурної	5.11.2019 р.	
8.	Розроблення схеми функціональної	10.11.2019 р.	
9.	Розробка стартап – проекту	15.11.2019 р.	
10	Оформлення текстової документації	28.11.2019 р.	

Студент _____ Шлапак С.С.
(підпис) (ініціали, прізвище)

Науковий керівник дисертації _____ Корнієнко Б. Я.
(підпис) (ініціали, прізвище)

РЕФЕРАТ

Магістерська дисертація на здобуття ступеня магістру на тему «Система управління IoT мережею»: 107с., 30 рис., 41 табл., 9 додатки, 31 джерел.

Об'єкт дослідження - система управління IoT мережею.

Мета роботи - покращення захисту IoT мереж та приватної інформації користувача.

У магістерській дисертації розглядається проблема захисту IoT мереж та приватної інформації користувача від сторонніх лиць. Пропонується концептуальний підхід для вирішення проблеми захисту з використанням існуючих стандартів захисту та їх модифікації. Особливістю системи є можливість самому обирати стандарт захисту, а також можливістьзбору та аналізу метрик в реальному часі.

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, СИСТЕМА УПРАВЛІННЯ ІОТ МЕРЕЖЕЮ, БАЗА ДАНИХ, КІБЕРЗАХИСТ

Master's thesis for master's degree on the topic "IoT network management system": 107p, 30 figures, 41 tables, 9 annexes, 31 sources.

The object of study is the IoT network management system.

The purpose of the work is to improve the protection of IoT networks and user private information.

The Master's thesis deals with the problem of protecting IoT networks and private user information from third parties. A conceptual approach is proposed to solve the security problem using existing security standards and their modifications. The peculiarity of the system is the ability to choose the standard of protection, as well as the ability to collect and analyze metrics in real time.

SOFTWARE, IOT NETWORK MANAGEMENT SYSTEM, DATABASE, CYBER PROTECTION

ЗМІСТ

ПЕРЕЛІК ВИКОРИСТАНИХ СКОРОЧЕНЬ 7
ВСТУП 9
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 11
2 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ 16
2.1 Asure Stream 16
2.2 Google Cloud IoT 18
2.3 AWS IoT Platform 20
2.4 ThingWorx – MDM IoT Platform 21
2.5 Bosch IoT Suite - MDM IoT Platform 23
3 ТЕОРЕТИЧНИЙ ОПИС РІШЕННЯ 27
3.1 Розробка вимог до системи 27
3.2 Загальний підхід до поставленої задачі 27
3.2.1 Захист мережі та інформації 28
3.2.2 Автоматизована конфігурація мережі 29
3.2.3 Збір даних та метрик 30
3.2.4 Протоколи комунікації 30
3.2.5 Керування мережею 34
3.3 Висновки 34
4 РЕАЛІЗАЦІЯ ТЕХНОЛОГІЧНОГО СТЕКУ 35
4.1 Архітектурне проектування програмного забезпечення 35
4.2 Вибір бекенд технологій 37
4.3 Вибір фронтенд технологій 38
4.3.1 Strongswan 39
4.3.2 Wpa-supplicant 41
4.3.2 Freeradius 42
4.4 Вибір сховищ даних 44
4.4.1 Реляційні бази даних SQL 44
4.4.2 Нереляційні бази даних NoSQL 45
4.4.3 Бекенд сховища даних 46
4.4.4 Фронтенд сховища даних 49

5. РОЗРОБКА СИСТЕМИ 52
5.1 Сценарії використання системи 52
5.2 Розробка баз даних 73
5.3 Опис концептуальної моделі даних 73
5.4 Розробка демону автоматичного конфігурування системи 75
5.4.1 Налаштування Dhclient 76
5.4.2 Налаштування wpa_supplicant 77
5.4.3 Налаштування Freeradius 78
5.5 Розробка збору метрик 79
5.6 Розробка користувацького інтерфейсу 83
5.7 Розробка користувацького інтерфейсу 87
6. РОЗРОБЛЕННЯ СТАРТАП ПРОЕКТУ 89
6.1 Опис ідеї проекту 89
6.2 Аналіз потенційних техніко-економічних переваг 90
6.3 Технологічний аудит ідеї проекту 92
6.4 Аналіз ринкових можливостей запуску стартап-проекту 94
6.5 Визначення потенційних груп клієнтів 95
6.6 Аналіз ринкового середовища 96
6.7 Аналіз пропозиції 99
6.8 Перелік факторів конкурентоспроможності 102
6.9 Аналіз сильних та слабких сторін стартап-проекту 103
6.10 SWOT-аналіз 104
6.11 Опис цільових груп потенційних клієнтів 105
6.12 Формування базових стратегій розвитку 105
6.13 Вибір стратегії конкурентної поведінки 106
ВИСНОВКИ 109
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ 111
ДОДАТКИ 114
Додаток А – Публікація 114

ПЕРЕЛІК ВИКОРИСТАНИХ СКОРОЧЕНЬ

Спринт – відрізок часу протягом якого розробляється чи випускається інкремент продукту.

СУБД – система управління базами даних

TLV – type,length,value.

Метрика – міра, що дозволяє оцінити числове значення властивостей програмного забезпечення.

TLS – криптографічний протокол по передачі даних в мережі Інтернет.

CA – субєкт , що видає криптографічні цифрові сертифікати.

ISO – міжнародна організація, що відповідає за ратифікацію розроблених спільно делегатами з різних країн стандартів.

CLI – command line interface

UI – засіб взаємодії користувача з системою.

X.509 - криптографічний стандарт для інфраструктури приватного ключа та керування привілегіями.

EAP – робота кадру аутентифікації, яка часто використовується в мережевих і Інтернет-з'єднаннях

CRL – список сертифікатів, які засвідчує центр позначив як не валідні.

RADIUS – це протокол передачі даних, який використовує компютерна мережа для проведення процедури автентикації

AAA – загальна назва процесів, пов'язаних із забезпеченням захисту даних в інформаційних системах, включаючи забезпечення аутентифікації, авторизації та аудиту, але без забезпечення доступності даних.

ACID – atomicity, consistency, isolation, durability

API - application programming interface

IEEE – це група стандартів для комп’юторних мереж

WPA – протокол безпеки для захисту бездротової компютерної мережі

TCP/IP – мережева модель передачі даних

DHCP – мережевий протокол для автоматичного розподілу IP адресс в підмережі

SSL – криптографічний протокол, призначений для забезпечення захисту зв'язку через комп'ютерну мережу

ВСТУП

Данна магістерська дисертація обумовлена бурхливим розвитком технології інтернет речей та потребою об’єднувати фізичні об’єкти, що пов’язані між собою за допомогою вбудованих технологій, для взаємодії між собою та зовнішнім світом, в одну захищену підмережу. Проблема захисту топології мережі та інформації користувача в мережах інтернет речей є актуальною [1]. Саме тому вибрана проблематика дослідження є актуальною в сучасних реаліях. Також потрібно підняти проблему маштабування, комунікація між розумними девайсами реалізована по різному від вендору до вендору. У більшості існуючих рішень можна зустріти зазначені проблеми, які тільки починають вирішуватись.

Метою даної магістрської дисертації є покращення захисту IoT мереж та приватної інформації користувача. Проаналізувати існуючі платформи та системи, запропонувати новий метод вибору стандарту захисту, а також алгоритм підбору протоколів комунікації в залежності від можливостей кожного окремого девайсу.

Відповідно до мети потрібно вирішити наступні завдання:

- розглянути стандарти та методи захисту мережі
- розробити алгоритм аналізу та підбору протоколу транспорту інформації
- описати реалізацію запропонованих рішень
- розробити тестову модель та топологію мережі

Обєкт дослідження – платформи для конфігурації та розгортування мереж інтернет речей.

Предметом дослідження є методи захисту мережі, та забезпечення стійкості системи.

В рамках магістерської дисертації дослідження предметної області, інструментів, проблем проводилось з використанням загальнонаукових та спеціальних методів, а саме:

- метод моделювання – розробка концепції на ранніх стадіях створення системи;
- аналіз та синтез – проведення дослідження існуючих варіантів захисту мережі, створення нових рішень та розробка програмного комплексу з використанням

високотехнічних засобів;

– метод збору метрик– модифікація системи для поліпшення її ефективності за рахунок збору, аналізу та часткового збереження даних.

– метод порівняння – огляд схожих варіантів реалізації захисту мережі, порівняння різних стандартів захисту, що дало змогу краще дослідити предметну область та розібратись в технічних вимогах до програмного комплексу.

Основним здобутком дисертації є оптимізація захищеності системи основана на стандарті захисту macsec, привязування даних до mac адресу та методи збору та аналізу інформації з девайсів, що підключені до однієї мережі. Все це дозволяє пободувати захищену та керовану мережу з меншими зусиллями, а також придає впевненості щодо захисту приватної інформації користувача. Дана система може бути використана як інструмент для розгортування та керування IoT мережею.

Отримана в результаті виконаної дослідницької роботи методологія оптимізації захисту мережі були опубліковані у статті «Порівняння стандартів безпеки комп'ютерних мереж» в сучасному науковому журналі «Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення» в 2019 році.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Інтернет речей – досить нова тема технічного, соціального та економічного значення. Споживчі товари, товари тривалого користування, автомобілі та вантажівки, промислові та комунальні компоненти, датчики та інші побутові предмети поєднуються з підключенням до Інтернету та потужними можливостями аналітики даних, які обіцяють перетворити спосіб роботи, життя та гри.

Прогнози щодо впливу IOT на Інтернет та економіку вражають, оскільки деякі очікують, що до 100-ти мільярдів підключених пристроїв IoT і глобального економічного впливу до 2025 року більше, ніж 11 мільйонів доларів [2].

Водночас, технологія інтернет речей викликає значні проблеми, які можуть стати на шляху реалізації його потенційних вигод. Привабливі заголовки про злому підключених до Інтернету пристроїв, проблеми спостереження та побоювання щодо конфіденційності вже привернули увагу громадськості. Технічні виклики залишаються, а нові політичні, правові та проблеми розвитку виникають.

Основні поняття, які служать основою для вивчення можливостей та викликів IoT, включають:

– Визначення IoT: Термін Інтернет речей, як правило, відноситься до сценаріїв, коли підключення до мережі та обчислювальна спроможність поширюється на об'єкти, датчики та предмети побуту, які зазвичай не вважаються комп'ютерами, дозволяючи цим пристроям генерувати, обмінюватися та споживати дані з мінімальним втручанням людини. Однак немає єдиного універсального визначення. На рисунку 1.1 зображено прогноз по монетизації та глобальний економічний вплив технології інтернет речай на окремі галузі.

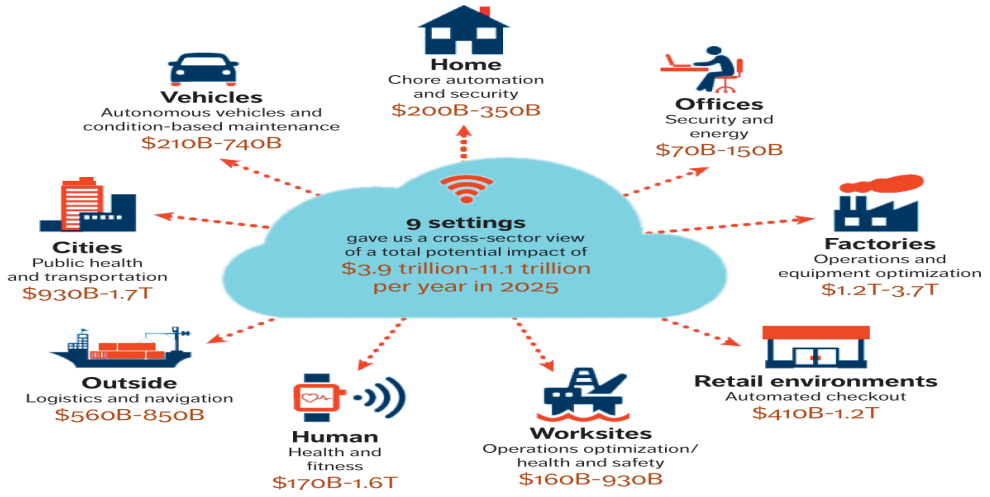

Рисунок 1.1 – Прогноз впливу на економіку окремих галузей [3]

– Включення технологій: Концепція поєднання комп'ютерів, датчиків та мереж для контролю та управління пристроями існує десятиліттями. Однак недавнє злиття декількох тенденцій на ринку технологій наближає Інтернет речей до широко розповсюдженої реальності. Сюди входять повсюдне підключення, широке прийняття мереж на основі IP, економіка обчислень, мініатюризація, досягнення в аналітиці даних та зростання хмарних обчислень. На рисунку 1.2 зображено сфери діяльності людини в яких технологія інтернет речей має місце бути.

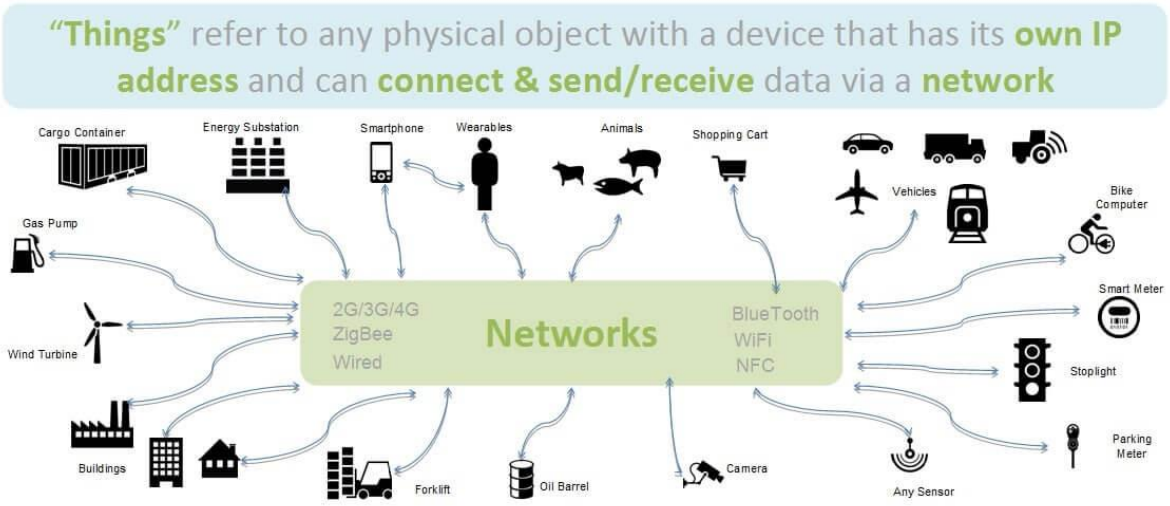

Рисунок 1.2 – Сфери взаємодії інтернет речами [4]

– Моделі підключення: В реалізаціях IoT використовуються різні моделі технічних комунікацій, кожна зі своїми характеристиками. Чотири поширені моделі комунікацій, описані Інститутом архітектури Інтернету, включають: "Пристрій до пристрою", "Пристрій до хмари", "Пристрій до шлюзу" та "Обмін даними на зворотній основі". Ці моделі підкреслюють гнучкість у способах підключення пристроїв IoT та надання користувачеві цінності. На рисунку 1.3 зображена загальна модель підключення та комунікації.

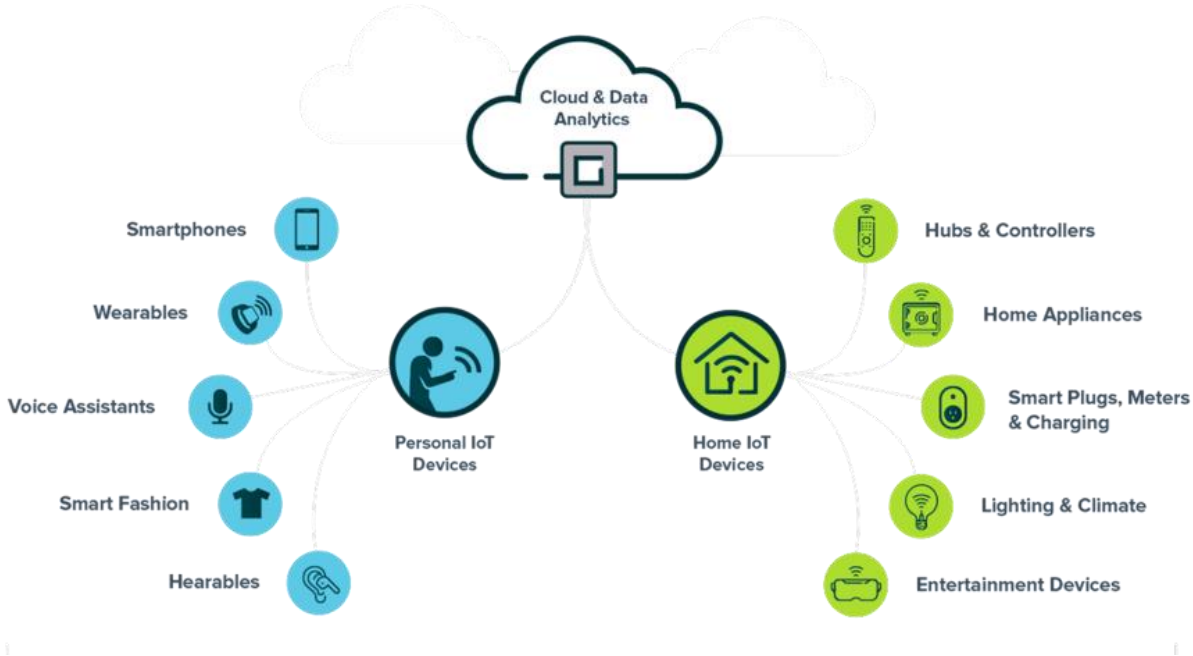

Рисунок 1.3 – Модель підключення та комунікації [5]

– Трансформаційний потенціал: Якщо прогнози та тенденції до IoT стануть реальністю, це може змусити змінитись у думках про наслідки та проблеми у світі, де найпоширеніша взаємодія з Інтернетом пов'язана з пасивною взаємодією з об'єднаними об'єктами, а не з активною взаємодією зі змістом. На рисунку 1.4 зображено потенціал росту популярності технології на найближчі роки. Потенційна реалізація цього результату - «гіперпов’язаний світ» - свідчить про загальний характер самої архітектури Інтернету, який не встановлює властивих обмежень програм або служб, які можуть використовувати цю технологію.

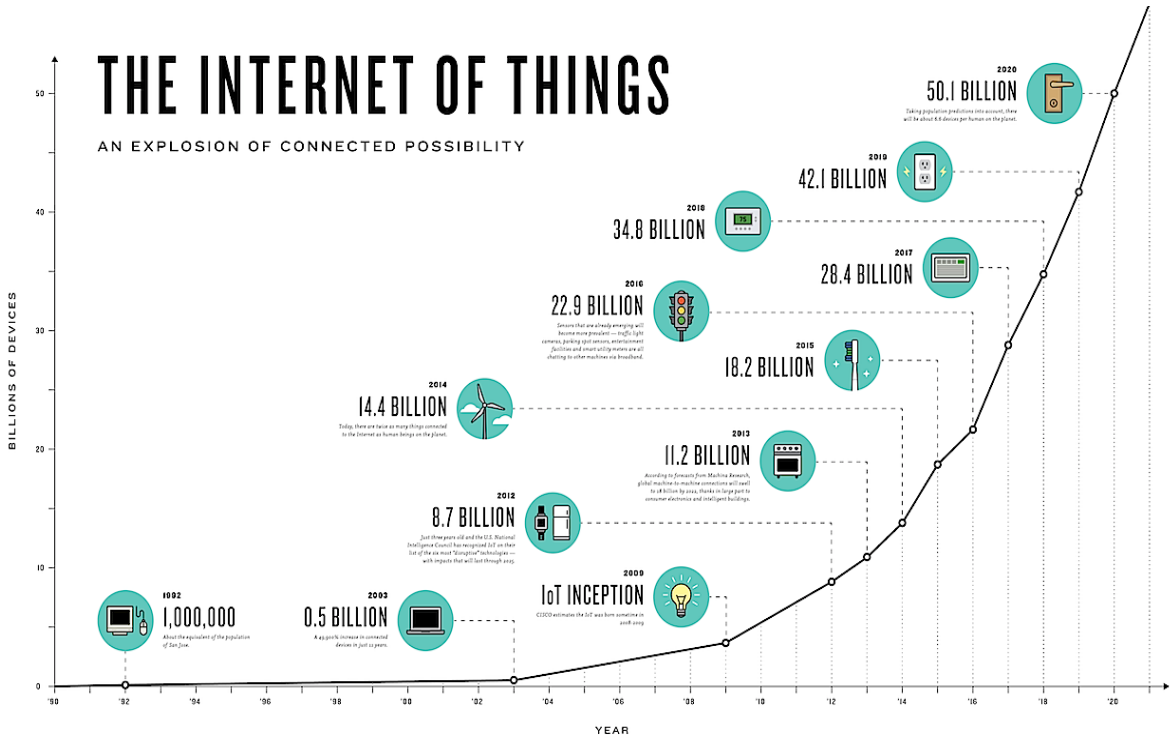

Рисунок 1.4 – Потенціал росту популярності технології [6]

Для вивчення деяких найактуальніших проблем та питань, пов'язаних з технологією, вивчається п’ять ключових питань IOT. До них належить безпека; конфіденційність; сумісність та стандарти; правові, регуляторні та права; та економіки, що розвиваються, та розвиток.

Розглянемо проблеми які притаманні даній технології:

– Безпека: Хоча міркування безпеки не є новими в контексті інформаційних технологій, атрибути багатьох реалізацій IoT представляють нові та унікальні проблеми безпеки. Вирішення цих завдань та забезпечення безпеки продуктів та послуг IoT повинно бути основним пріоритетом. Користувачі повинні довіряти, що пристрої IoT та пов'язані з ними послуги передачі даних захищені від вразливих ситуацій, тим більше, що ця технологія стає все більш поширеною та інтегрованою у наше повсякденне життя. Погано захищені пристрої та сервіси IoT можуть слугувати потенційними точками входу для кібератаки та піддавати користувачам крадіжки, залишаючи потоки даних недостатньо захищеними.

– Конфіденційність: Повний потенціал Інтернету речей залежить від стратегій, що враховують індивідуальний вибір конфіденційності в широкому спектрі

очікувань. Потоки даних та специфічність користувачів, що надаються пристроями IoT, можуть розблокувати неймовірну та унікальну цінність для користувачів IoT, однак занепокоєння щодо конфіденційності та потенційної шкоди може стримувати повне прийняття Інтернету речей. Це означає, що права на конфіденційність та повага до очікувань щодо конфіденційності користувачів є невід'ємною частиною забезпечення довіри та довіри користувачів до Інтернету, підключених пристроїв та пов'язаних з ними послуг.

– Взаємодія / Стандарти: Роздроблене середовище власних технічних реалізацій IoT пригнічуватиме значення для користувачів та галузі. Хоча повна сумісність між продуктами та послугами не завжди є здійсненною або необхідною, покупці можуть не вагатися купувати продукти та послуги IoT, якщо є інтегральна негнучкість, висока складність власності та занепокоєння з приводу блокування постачальників.

– Нові проблеми економіки та розвитку: Інтернет речей має велику обіцянку для надання соціальної та економічної вигоди економікам, що розвиваються та розвиваються. Сюди входять такі сфери, як стійке сільське господарство, якість та використання води, охорона здоров'я, індустріалізація та управління довкіллям. Таким чином, IoT обіцяє як інструмент досягнення цілей ООН з питань сталого розвитку.

2 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

Платформа IoT - це багатошарова технологія, яка забезпечує можливість прямого забезпечення, управління та автоматизації підключених пристроїв Internet of Things. Він в основному підключає ваше обладнання, хоч і різноманітне, до хмари, використовуючи гнучкі параметри підключення, механізми безпеки підприємства та широкі повноваження щодо обробки даних. Для розробників платформа IoT пропонує набір готових до використання функцій, які значно прискорюють розробку програм для підключених пристроїв, а також дбають про масштабованість та сумісність між пристроями [7].

Для побудови мережі інтернет речей потрібна платформа, що дасть можливість обмінюватись даними між хостами, де в якості хосту можуть виступати не тільки комп’ютери, а й інші девайси з можливістю комунікації. Розглянемо платформи для створення своїх додатків та IoT мереж.

2.1 Asure Stream

Це аналітика в реальному часі і складна система обробки подій, яка призначена для аналізу та обробки великих обсягів швидкої потокової передачі даних з декількох джерел одночасно. Шаблони та взаємозв'язки можна визначити в інформації, отриманій з ряду вхідних джерел, включаючи пристрої, датчики, кліки, соціальні мережі та програми. Ці шаблони можуть бути використані для запуску дій та ініціювання робочих процесів, таких як створення сповіщень, подача інформації до інструмента звітування або зберігання перетворених даних для подальшого використання. Крім того, Stream Analytics доступний в режимі виконання Azure IoT Edge та підтримує таку ж точну мову чи синтаксис, що і хмара.

Особливості та можливості Azure Stream Analytics:

- аналіз потоків телеметрії в реальному часі від пристроїв IoT;
- веб-журнали (clickstream analytics);
- геопросторова аналітика для управління автопарком і без водіїв;

– віддалений моніторинг та прогнозне обслуговування високоцінних активів.

Завдання Analytics Azure Stream складається з введення, запиту та виводу. Потокова аналітика передає дані з концентраторів подій Azure, Azure IoT Hub або Azure Blob Storage. Запит, який базується на мові запитів SQL, може використовуватися для легкого фільтрування, сортування, агрегації та приєднання потокових даних протягом певного періоду часу. Також можна розширити цю мову SQL за допомогою визначених користувачем функцій JavaScript та C #.

Кожне завдання має вихід для перетворених даних, і можна контролювати, що відбувається у відповідь на інформацію, що аналізувалась. Також можна зберігати дані в інших службах зберігання Azure для підготовки моделі машинного навчання на основі історичних даних або проведення пакетної аналітики.

На рисунку 2.1 показано, як дані надсилаються в Stream Analytics, аналізуються та надсилаються для інших дій, таких як сховище.

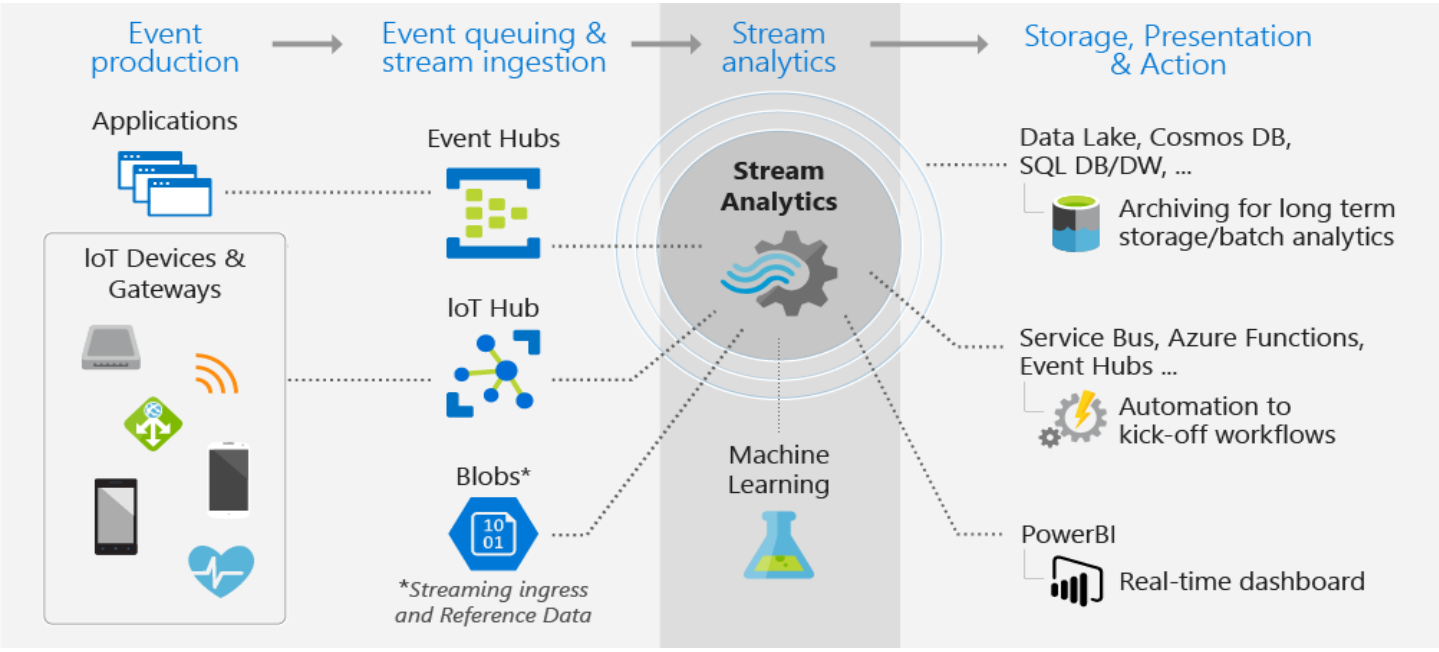

Рисунок 2.1 – Процес обміну диних в Stream Analytics [8]

Далі наведені переваги, недоліки Asure Stream.

Переваги:

– пропонує сторонні послуги;

– безпечний і масштабований;

– висока доступність.

Недоліки:

- вимагає управління;
- дорогий;
- немає підтримки для помилок.

2.2 Google Cloud IoT

Платформа для інтелектуальних послуг IoT Google Cloud IoT - це повний набір інструментів для підключення, обробки, зберігання та аналізу даних як на краю, так і в хмарі. Платформа складається з масштабованих, повністю керованих хмарних сервісів; інтегрований стек програмного забезпечення для кращих / локальних обчислень з можливостями машинного навчання для всіх ваших потреб IoT. Безпечне підключення та управління пристроєм Cloud IoT Core - це повністю керований сервіс, який дозволяє легко та безпечно підключатись, керувати та приймати дані з мільйонів глобально розповсюджених пристроїв. Cloud IoT Core у поєднанні з іншими сервісами на платформі Cloud IoT забезпечує повне рішення для збору, обробки, аналізу та візуалізації даних IoT в режимі реального часу для підтримки підвищеної операційної ефективності. Безпечно підключіть існуючу мережу пристроїв

Можливість безпечно підключити кілька або мільйони глобально дисперсних пристроїв через кінцеві точки протоколу, які використовують автоматичне балансування навантаження та горизонтальне масштабування, щоб забезпечити плавне введення даних при будь-яких умовах. Cloud IoT Core підтримує стандартні протоколи MQTT і HTTP, тому можна використовувати наявні пристрої з мінімальними змінами програмного забезпечення.

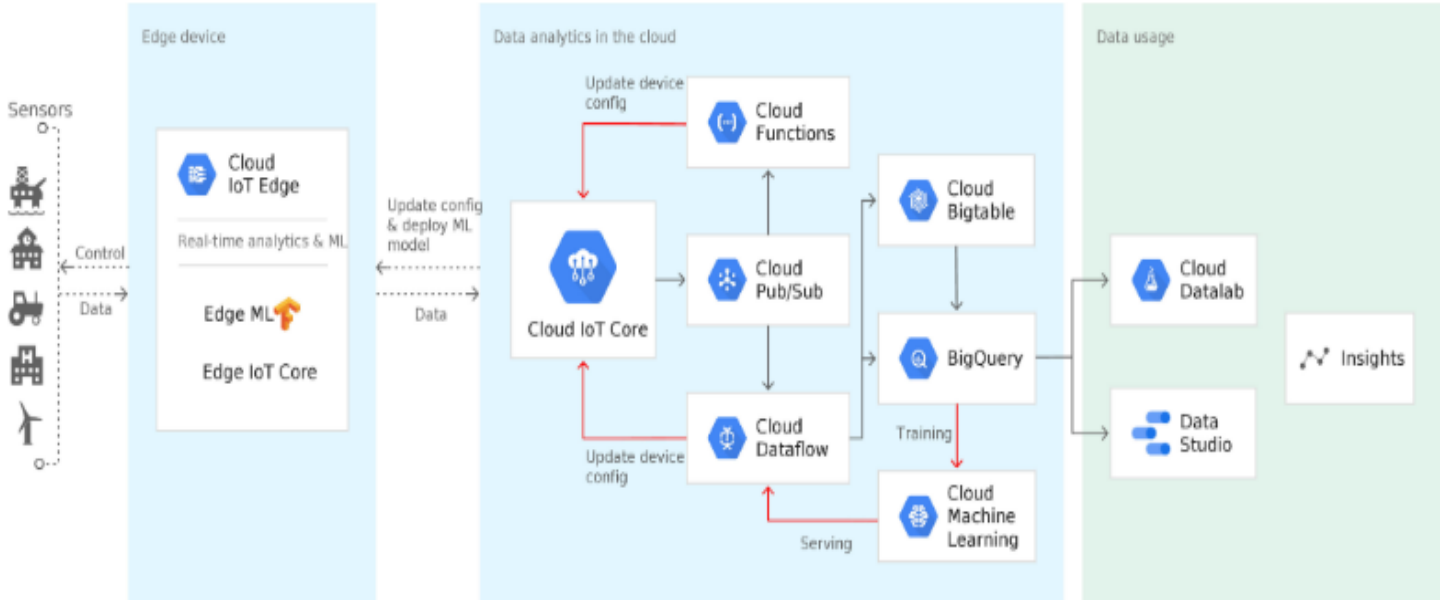

Рисунок 2.2 – Внутрішня інфраструктура Google Cloud IoT [9]

Cloud IoT Core працює на безсерверній інфраструктурі Google, яка автоматично масштабується у відповідь на зміни в режимі реального часу та дотримується суворих стандартних галузевих протоколів безпеки, що захищають дані вашого бізнесу. На рисунку 2.2 зображена внутрішня інфраструктура рішення.

Увімкнути захист від кінця до кінця, використовуючи асиметричну аутентифікацію ключа через TLS 1.2; Для підтвердження права власності на пристрій можна використовувати сертифікати, підписані CA. Пристрої, що підтримують вимоги безпеки Cloud IoT Core, можуть забезпечити повний стек безпеки. Сервіс використовує Cloud Pub / Sub під ним, який зберігає дані протягом семи днів.

Далі наведені переваги, недоліки Google Cloud IoT.

Переваги:

– найшвидший вхід / вихід;
– менший час доступу;
– забезпечує інтеграцію з іншими службами Google.

Недоліки:

– більшість компонентів - це технології Google;
– обмежений вибір мови програмування.

2.3 AWS IoT Platform

Платформа AWS IoT забезпечує підключення пристроїв до сервісів AWS і інших пристроїв, безпеку даних і взаємодії, обробку даних пристроїв і різні операції з ними, а також взаємодія додатків з пристроями навіть при відсутності підключення до Інтернету.

Платформа AWS IoT Core забезпечує взаємну аутентифікацію і шифрування в усіх точках підключення. Таким чином, будь-який обмін даними між пристроями і AWS IoT Core відбувається тільки після підтвердження ідентифікації. AWS IoT Core підтримує метод аутентифікації AWS, аутентифікацію на основі сертифікату X.509 і спеціальну аутентифікацію на основі токенів (через настроюються модулі авторизації). AWS IoT Core дозволяє використовувати сертифікати, сформовані самим сервісом AWS IoT Core, а також сертифікати, підписані обраним центром сертифікації. Існує можливість прив'язати до кожного сертифікату обрані політики для надання авторизованого доступу пристроїв або додатків з можливістю відкликання дозволу доступу без необхідності працювати з пристроєм безпосередньо. Можна створювати та використовувати сертифікати і політики пристроїв і керувати ними з консолі або за допомогою API. Ці сертифікати пристроїв можна надавати, активувати і пов'язувати з відповідними політиками IoT, налаштованими за допомогою AWS IoT Core. Така схема дозволяє при необхідності відразу ж відкликати дозвіл доступу для конкретного пристрою. AWS IoT Core також підтримує підключення з боку мобільних додатків користувачів за допомогою сервісу Amazon Cognito, який повністю забезпечує створення унікальних ідентифікаторів для користувачів ваших додатків і отримання тимчасових даних доступу AWS з обмеженими правами. AWS IoT Core також може надавати тимчасові дані для доступу AWS, після того як пристрій аутентифицироваться за допомогою сертифіката X.509. Це дозволяє пристрою простіше отримати доступ до інших сервісів AWS, таким як DynamoDB або S3. На рисунку 2.3 зображена внутрішня архітектура рішення.

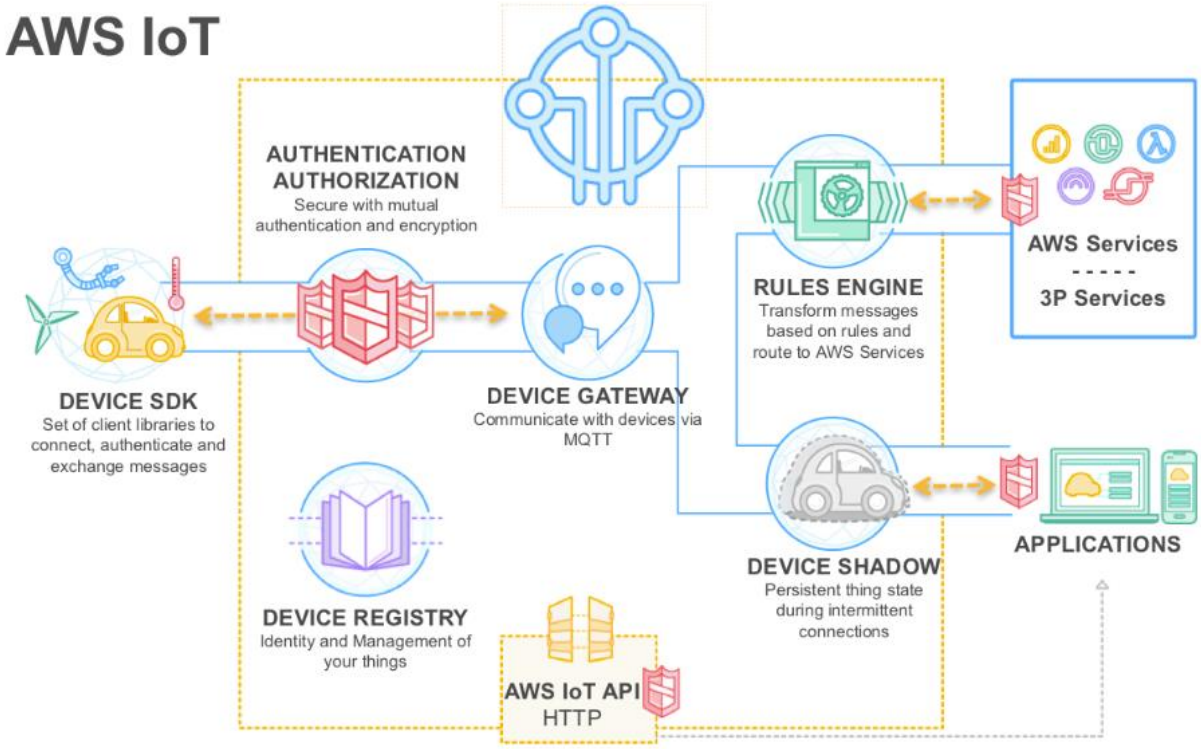

Рисунок 2.3 –Внутрішня архітектура AWS IoT Platform [10]

Далі наведені переваги, недоліки AWS IoT Platform.

Плюси:

– хороша інтеграція з пропозиціями Laas;
– ціна впала за останні шість років;
– відкритий і гнучкий.

Мінуси:

– велика крива навчання для AWS;
– три відключення за останні 2 роки;
– не безпечно для розміщення критичних корпоративних програм.

2.4 ThingWorx – MDM IoT Platform

ThingWorx є унікальною серед платформ IIoT, що пропонує найповніший набір найважливіших можливостей IIoT, як на самому, так і завдяки надійній інтеграції з такими

партнерами, як Microsoft та Rockwell. Це платформа для побудови промислового інтернету речей, побудована спеціально для застосувань промислового маштабу.

Маючи широкий досвід в галузі домену, побудований на основі майже двох десятиліть інновацій IoT, ThingWorx PTC - це платформа IIoT з функціональністю та гнучкістю, необхідною для швидкої рентабельності інвестицій - при цьому пропонуючи безпеку та масштабованість, необхідну для розширення рішень IIoT на всьому підприємстві.

Платформа управляє спілкуванням між системами, людьми та речами, а також обробляє перетворення даних, збереження даних та бізнес-логіку, щоб ви могли зосередитись на розробці програми. Автоматизація замість кодування вручну та підтримки прогнозних моделей. На рисунку 2.4 зображена структурна схема рішення ThingWorx, програмне рішення побудоване на основі модульної архітектури.

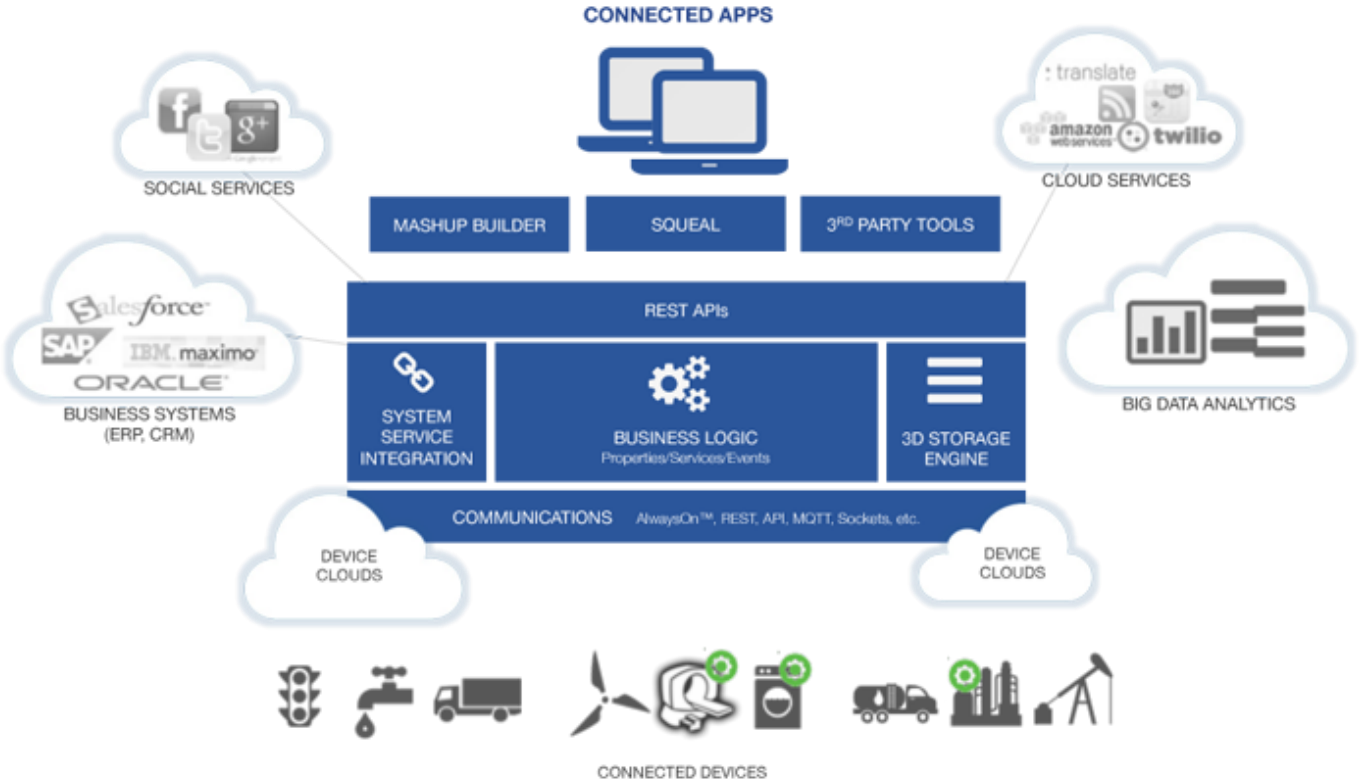

Рисунок 2.4 – Структурна схема ThingWorx [11]

ThingWorx інтеграція LDAP для користувачів та груп, дає можливість ділити девайси по групам користувачів, платформа надає швидке перетворення даних з ваших пристроїв у інформацію, яку легко зрозуміти в виді графіків, візуалізації та метрик.

ThingWorx пропонує найширший спектр драйверів доступні, підтримуючи поточні та застарілі пристрої, провідні та бездротові мережеві носії, і підключення до баз даних, програмного забезпечення додатків та інших серверів OPC.

Безпечні, автентифіковані та зашифрованікомунікації через різні мережі топології відповідають ряду стандартів, а також конфігурація захищених тунелів даних.

Далі наведені переваги, недоліки ThingWorx – MDM IoT Platform.

Плюси:

- хороша інтеграція з пропозиціями Laas;
- ціна впала за останні шість років;
- відкритий і гнучкий.

Мінуси:

- велика крива навчання для AWS;
- три відключення за останні 2 роки;
- не безпечно для розміщення критичних корпоративних програм.

2.5 Bosch IoT Suite - MDM IoT Platform

Bosch пропонує попередньо упаковані сервісні пропозиції для управління пристроями для масштабного розгортання: клієнтам надається велика частина зусиль із розробки, зазвичай необхідних для цілісного інтегрування різних послуг.

Служби Bosch IoT Remote Manager та Bosch IoT Rollouts забезпечують відмінне управління життєвим циклом в масштабі: спеціально орієнтовані на більші розгортання керованих пристроїв у корпоративних середовищах, комбінація Bosch IoT Remote Manager для управління життєвим циклом та Bosch IoT Rollouts для управління програмним забезпеченням та програмним забезпеченням забезпечує ефективне управління рішення для контролю над великою кількістю пристроїв.

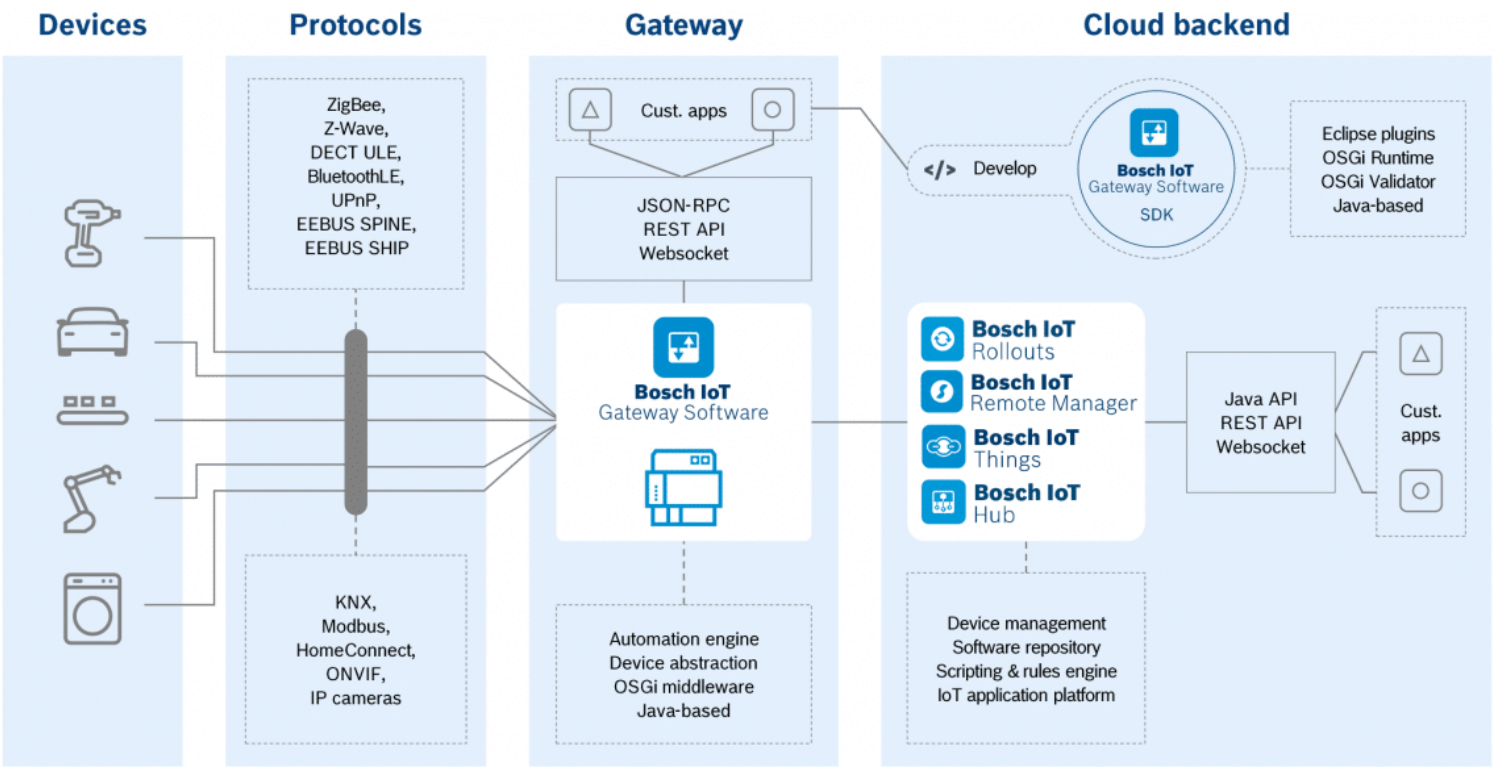

Рисунок 2.5 – Внутрішня інфраструктура Bosch IoT Suite [12]

Служби Bosch IoT Suite можна розгортати у вибраних клієнтом приватних хмарах, загальнодоступних хмарах та локальних мережах (Bosch IoT Remote Manager) гнучко та масштабовано: для клієнтів із більш обмеженими вимогами щодо безпеки даних ці можливості дозволяють рішення пристрою Bosch для управління пристроями. повністю працювати в системах, що належать клієнтам або контролюються клієнтами, одночасно дозволяючи гнучкості та простоті керованих пропозицій послуг для клієнтів, які бажають використовувати громадські хмарні платформи. На рисунку 2.5 продемонстровано внутрішню інфраструктура рішення.

Bosch забезпечує гнучку та розширювану інтеграцію протоколів на південний пристрій: рівень адаптерів протоколу забезпечує інтеграцію з широким спектром протоколів на південь, включаючи типові протоколи IoT, такі як MQTT і CoAP, стандарти управління пристроями, такі як LWM2M і TR-069, і різноманітні галузеві стандарти власні протоколи.

Далі наведені переваги, недоліки Bosch IoT Suite - MDM IoT Platform.

Переваги:

- хороша інтеграція з пропозиціями Laas;
- ціна впала за останні шість років;
- відкритий і гнучкий.

Недоліки:

- велика крива навчання для AWS;
- три відключення за останні 2 роки;
- не безпечно для розміщення критичних корпоративних програм.

2.6 Висновоки

У розділі було проведено огляд та аналіз існуючих систем керування мережами інтернет речей, їх класифікацію за різними показниками, а також виділено їх сильні та слабкі сторони. В таблиці 2.1 представлено порівняльну характеристику існуючих рішень, на основі проаналізованої інформації.

Таблиця 2.1 – Порівняльна характеристика існуючих рішень

IoT платформа	Управління девайсами	Методи інтеграція	Методи захисту	Протоколи комунікації	Підтримка візуалізації
Asure Stream	Так	REST API	Authentication (X.509)		
Google Cloud IoT	Так	REST API			Yes

AWS IoT	Так	REST API	Link Encryption (TLS), Authentication (SigV4, X.509)	MQTT, HTTP1.1	Yes (AWS IoT Dashboard)
ThingWorx	Так	REST API	Standards (ISO 27001), Identity Management (LDAP)	MQTT, AMQP, XMPP, CoAP, DDS, WebSockets	Yes (ThingWorx SQUEAL)
Bosch IoT	Так	REST API	Unknown	MQTT, CoAP, AMQP,STOMP	Yes (User Interface Integrator)

Отже, опираючись на проаналізовану інформацію, що була продемонстрована в таблиці, можна виділити декілька основних конкурентів.

3 ТЕОРЕТИЧНИЙ ОПИС РІШЕННЯ

3.1 Розробка вимог до системи

Система розробляється як платформа для розгортування, керування, моніторингу та аналізу мережей інтернет речей. Функціональні вимоги включають в себе можливість підбору найкращого з допустимих стандартів захисту мережі та інформації користувача, вибір необхідного протоколу комунікації між девайсами.
Збір даних та необхідних метрик з девайсів, що обєднані в одну мережу повинна бути візуалізована та наглядна. Конфігурування мережі повинна проходити в автоматичному режимі, а у випадку коли користувач обирає режим адміністратора система повинна проводити конфігурацію в напівавтоматичному режимі. Керування мережею повинно здійснюватись за допомогою додатку, веб-застосунку або командною строкою.

Список функціональних вимог до системи:

- збір даних та метрик;
- керування мережею;
- зрозумілий користувацький інтерфейс.

Список нефункціональних вимог до системи:

- автоматизована конфігурація мережі;
- захист мережі та інформації;
- автоматизований підбір необхідного протоколу комунікації.

3.2 Загальний підхід до поставленої задачі

Нижче розглянуто та проаналізовано кожну з функціональних та нефункціональних вимог, а також обрано методи та підходи їх вирішення.

3.2.1 Захист мережі та інформації

Як специфікація другого рівня, MACsec може захищати не тільки IP-трафік, але й ARP, відкриття сусіда та DHCP. Він покладається на GCM-AES для забезпечення конфіденційності та цілісності всього мережевого трафіку.

MACsec був стандартизований у 2006 році IEEE (стандарт IEEE 802.1AE-2006), але підтримка була додана лише нещодавно до основного ядра Linux (станом на 4.6). У MACsec пакети перетікають через «захищені канали», які підтримуються «захищеними асоціаціями». Кожна захищена асоціація використовує окремий, випадковим чином генерований ключ [13].

IEEE 802.1X-2010 визначає супровідний протокол, MACsec Key Agreement (MKA), який забезпечує обмін ключами та дозволяє взаємну аутентифікацію вузлів, які хочуть взяти участь у асоціації з'єднань MACsec. На машинах Linux це реалізовано в wpa_supplicant. wpa_supplicant використовує деякий маркер автентифікації (загальнодоступний ключ або пару імен користувача / пароля, аналогічно механізмам аутентифікації, які використовуються в WiFi) для встановлення сеансу з сервером аутентифікації, який може бути комутатором або хостом Linux, що працює з hostapd. Після аутентифікації ключі генеруються та обмінюються (по зашифрованому каналу) і використовуються для налаштування захищеного зв'язку MACsec.

На дроті MACsec-пакет починається із заголовка Ethernet з EtherType 88E5. Далі йде MACsec SecTAG, який містить інформацію, яка допомагає одержувачу ідентифікувати ключ дешифрування, а також номер пакету (для захисту від повторного відтворення). Після SecTAG приходить корисне навантаження, яке можна зашифрувати, і ICV (Integrity Check Value), яке генерується GCM-AES, і гарантує, що пакет справді був створений вузлом, у якому був ключ, і hasn не змінено в дорозі [14].

Перш за все, MACsec і IPsec працюють на різних мережевих шарах. IPsec працює над IP-пакетами, на рівні 3, тоді як MACsec працює на рівні 2, на кадрах Ethernet. Таким чином, MACsec може захищати весь трафік DHCP та ARP, який IPsec не може захистити. З іншого боку, IPsec може працювати через маршрутизатори, тоді як MACsec обмежений локальною мережею. Як MACsec, так і IPsec, користувацькі

програми не потребують змін, щоб скористатися гарантіями безпеки, що надаються цими стандартами. У Red Hat Enterprise Linux підтримка IPsec надається пакетом libreswan [15].

SSL / TLS працює на ще одному шарі, а саме на п’ятому (додатковому) шарі. Таким чином, безпека програм може вимагати змін програми, що може виявитися досить проблематичним.

З іншого боку, вставлення криптографічного шару безпосередньо у додатки також пропонує деякі переваги. Програмне забезпечення може самостійно перевіряти стан шифрування та справжність на основі політики і може відповідно реагувати. Перевірка цих властивостей безпеки безпосередньо в додатку може забезпечити захист від кінця до кінця.

У Red Hat Enterprise Linux SSL / TLS надається декількома бібліотеками, такими як GnuTLS, NSS, OpenJDK / JSSE і OpenSSL (в алфавітному порядку).

3.2.2 Автоматизована конфігурація мережі

Для того, щоб мережа функціонувала так як повинна, потрібно заповнити конфігураційні файли для dhclient та supplicant. Перший потрібен для того, щоб надати кожному з наших вузлів свою ip адресу, а інший відповідає за стандарти захисту інформації, котрі були розглянуті вище.

Вносити зміни до конфігураційних файлів при додавані нового вузлу чи частковій зміні топології мережі інтернет речей вимагає багато людських ресурсів для адміністрування. Саме для цього було розроблено демон, що працює в фоновому режимі, а також автоматично змінює конфігурацію. За підключенням та трафіком спостерігає dhclient демон, а також час від часу відсилає тестові пакети для заповнення таблиці маршрутизації за допомогою широкомовної передачі. В момент коли додається новий девайс, на нього приходить тестовий пакет від демону, котрий знаходиться на dhcp сервері. Таким чином за допомогою dhcp hook’у розроблений мною демон, що працює в фоновому реживі розуміє, коли потрібно змінитювати конфіг файли.

3.2.3 Збір даних та метрик

Аналіз та моніторинг інформації зібраної з девайсів підключених до однієї мережі є однією з основних вимог при розробці системи управління IoT девайсами. Користувач повинен знати, що відбувається з його девайсами в кожен момент часу. Для вирішення даної задачі потрібно розробити протокол, що дозволить запаковувати потрібну нам інформацію та відправляти її за допомогою протоколу комунікації між девайсами. Для цього портібнно описати так звані TLV. TLV – тип, довжина та значення. Також потрібно розробити механізм, який на основі інформації отриманої з наших кастомних метрик зможе побудувати графіки та візуалізувати всю отриману інформацію.

3.2.4 Протоколи комунікації

Передача даних між девайсами та вузлами проходить за допомогою протоколів потокової передачі даних. Розглянемо декілька найбільш популярні із них нижче:

1) wifi є очевидним вибором для підключення до IoT, що знаходиться в теперішньому часі покриттям Wi-Fi у будівництві практично практичне, але одночасно це не завжди є правильним. Розглянемо переваги та недоліки wifi нижче.

Переваги:

- Низька вартість інфраструктури та пристроїв
- Легкість розгортання
- Точки присутності

Недоліки:

- Високе енергоспоживання
- Помірний діапазон
- Скупчення спектра

Стандартний WiFi, хоча є очевидним вибором для IoT, має обмеження як у діапазоні, так і в енергоефективності, але деякі програми IoT можуть

використовувати встановлений стандартний WiFi, особливо для приміщень, що знаходяться в будівлі чи кампусі. Очевидними є випадки автоматизації будівництва та дому та енергоменеджменту, де встановлений WiFi можна використовувати як канал зв'язку, а пристрої можна підключати до електричних розеток.

Wi-Fi продовжують вдосконалювати IoT, а деякі починають надавати корпоративним точкам доступу за допомогою технологій IoT (ZigBee, Bluetooth та / або Thread);

2) Правильно розроблені додатки Bluetooth IoT ефективні для таких сценаріїв відстеження активів у приміщенні, враховуючи низькі енергоспоживання, теоретично нескінченну масштабованість та надійність самовідновлення нових мережевих мереж Bluetooth.Технологія Bluetooth може бути використана в декількох видах топологій, таких як пара, трансляція та сітка (рисунок 3.1).

Рисунок 3.1 –Види топологій технології Bluetooth [16]

Види підключення:

– один до одного: Bluetooth як засіб з'єднання двох пристроїв;

– один до багатьох: Bluetooth - це засіб передачі інформації на один пристрій багатьом пристроям або навпаки;

– багато-багато-багато: Bluetooth - це спосіб підключення багатьох пристроїв до багатьох інших, як ніби в павутинній павутин.

3) Зв'язок Zigbee спеціально побудований для мереж управління та датчиків за стандартом IEEE 802.15.4 для бездротових персональних мереж (WPAN), і це продукт від альянсу Zigbee. Цей стандарт зв’язку визначає фізичні та контрольні рівні доступу до засобів масової інформації (MAC) для обробки багатьох пристроїв із низькою швидкістю передачі даних [17]. Zigbee - це недорога і малопотужна сітчаста мережа, широко розгорнута для контролю та моніторингу програм, де вона охоплює 10-100 метрів у межах дальності. Ця система зв'язку дешевша і простіша за інші фірмові бездротові сенсорні мережі, як Bluetooth і Wi-Fi.

Zigbee підтримує декілька мережевих топологій; однак найбільш часто використовувані конфігурації - це зірки, сітки та топології дерев кластера (рисунок 3.2).

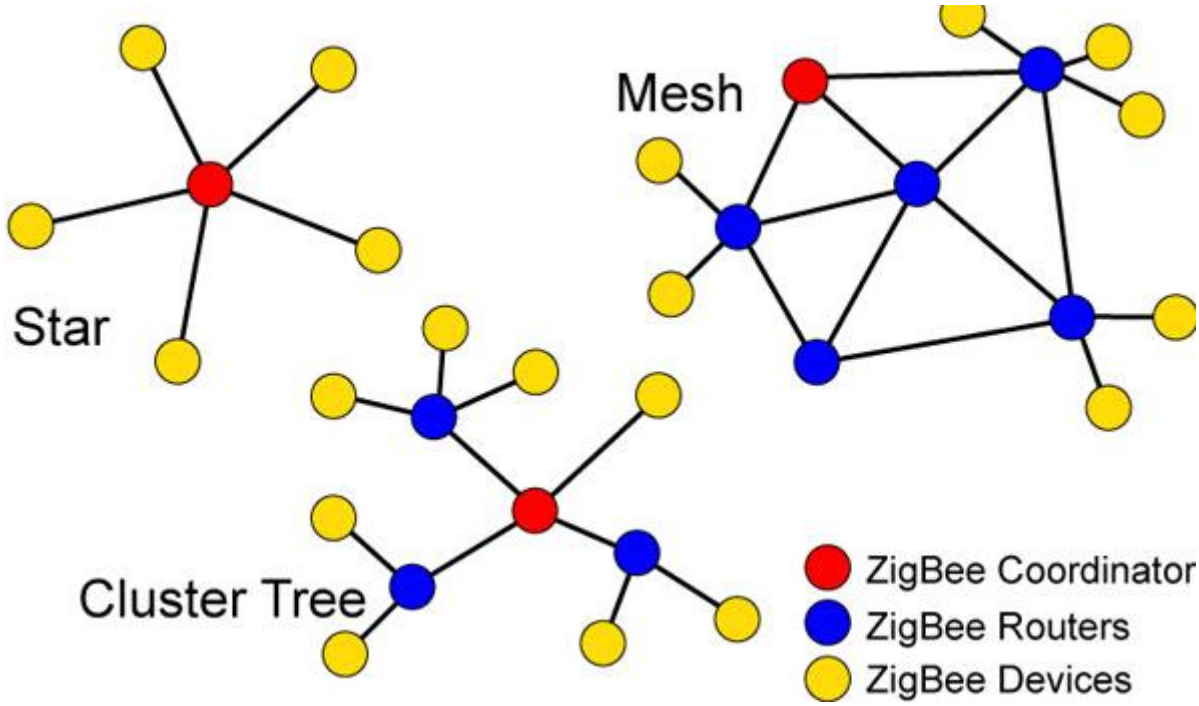

Рисунок 3.2 –Мережеві топології Zigbee [18]

Завдяки таким перевагам технології Zigbee, як низька вартість та низькі режими роботи енергії та її топології, ця технологія зв'язку невеликого діапазону найкраще підходить для декількох застосувань порівняно з іншими власними комунікаціями, такими як Bluetooth, Wi-Fi тощо.

Таблиця 5.1 – Порівняльна характеристика технологій

	ZigBee	Wi-Fi	Bluetooth
Стандарт фізичного рівня	802.15.4	802.11	802.15.1
Енергоживлення	низьке	високе	середнє
Пропускна здатність	20 – 250 kbit/s	11,000+	720
Радіус дії (м)	1-100+	1-30+	1-10+
Архітектура мережі	сітка	зірка	зірка
Переваги	низьке споживання електроенергії, дешевизна, маштабування, рентабільність	швидкість	Дешевизна, зручність
Розроблено для	Моніторингу та контролю	Веб застосунків та відео трафіку	Заміна кабелю, потокове аудіо

Отже, розглянувши основні технології передачі пакетів даних та проаналізувавши їх характеристики наведені в таблиці 1 можна зробити висновок, що навіть коли у девайса реалізовані всі розглянуті технології, нам потрібно обирати найкращу з них, для оптимізації роботи. Саме для цього потрібно розробити скрипт чи утіліту, що зможе опираючись на те, що потрібно в даний момент виконати буде обирати найбільш оптимальне рішення.

3.2.5 Керування мережею

Інтерфейс взаємодії з мережею є одним з ключових завдань, користувач повинен бути впевнений в кожен момент часу, що його система працює справно, а у випадку неконтрольваних випадків, система повинна повідомити користувача про неполадки, а також надати йому можливість впливати на робочий процес системи. Також потрібно надати можливість адміністратору мережі можливість вносити правки в конфігурацію, а також додавати нові вузли та змінювати топологію мережі. Саме для вирішення цих задач потрібно розробити зрозумілий користувацький інтерфейс та інтерфейс оператора для адміністрування.

Вимоги до користувацького інтерфейсу:

- простота використання;
- інформативність;
- візуалізація процесу.

Вимоги до інтерфейсу адміністратора:

- доступ до конфігураційних файлів;
- можливість моніторингу трафіку;
- зрозумілий інтерфейс;
- широкий спектр команд налаштування.

3.3 Висновки

У розділі було детально розглянуто та проаналізовано основні функціональні та не функціональні вимоги до системи, а також описано які задачі виконує кожна з них. З проаналізованих даних, можна зробити висновок, що основними вимогами до систем керування мережами інтернет речей є надійний захист мережі та приватної інформації користувача, можливість збору та аналізу даних, а також можливість керування та моніторингу системи в реальному часі.

4 РЕАЛІЗАЦІЯ ТЕХНОЛОГІЧНОГО СТЕКУ

4.1 Архітектурне проектування програмного забезпечення

Для реалізації системи було обрано модульну архітектуру. На рисунку 4.1 показана архітектура розроблюваної системи.

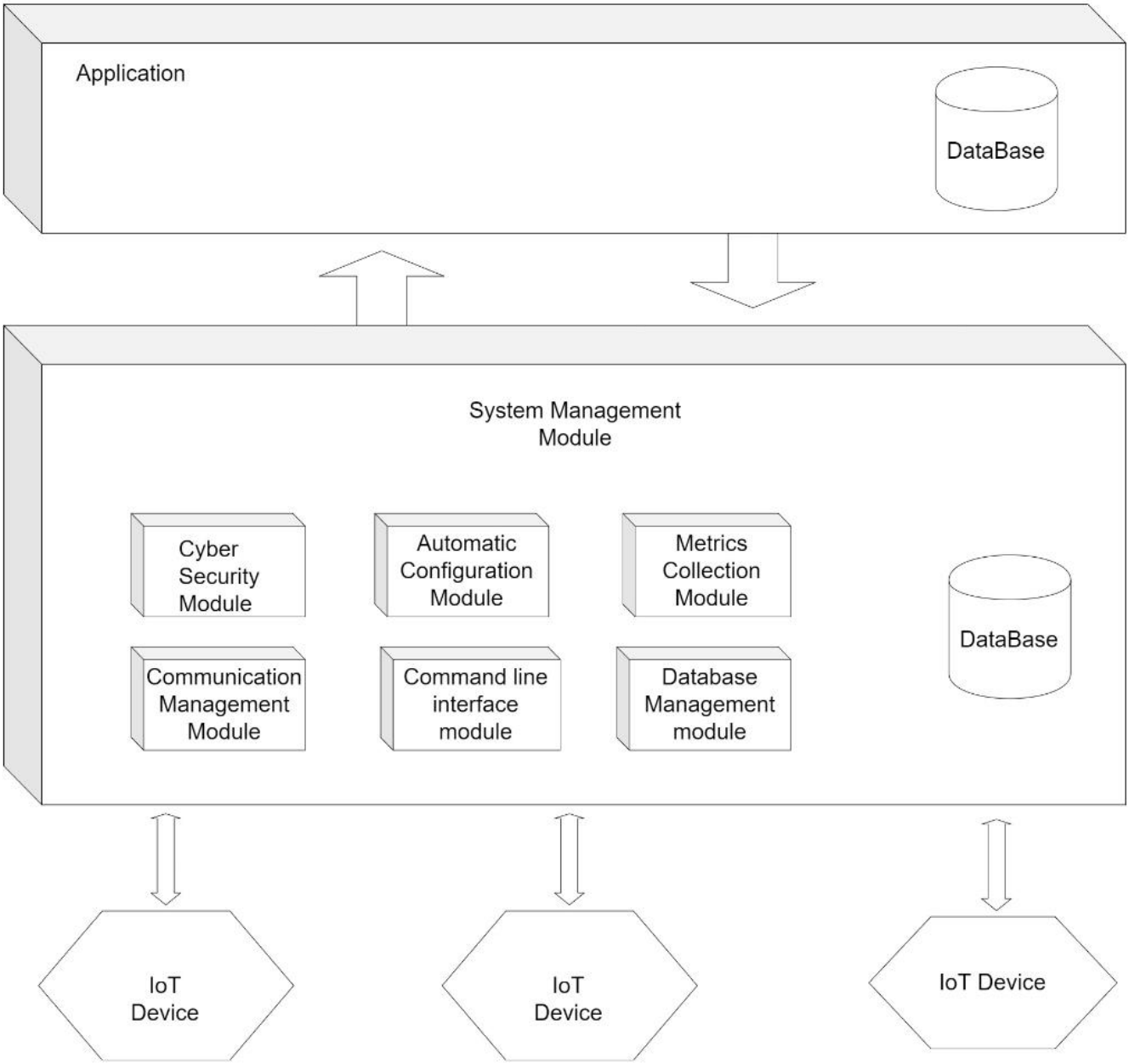

Рисунок 4.1 –Архітектура системи.

Архітектура поділяється на три основних шари:

- application layer;
- system management layer;
- iot devices layer.

Розглянемо кожен шар починаючи з верхнього, котрий називається application layer. Це рівень користувацького інтерфейсу, на ньому відбувається візуалізація того, що відбувається в даний момент часу з нашою системою.

Наступний шар system management layer було розбито на декілька програмних модулів, а саме :

- кібербезпеки: даний модуль повністю відповідає за забезпечення захисту інформації користувача та зихещеності мережі в цілому;
- комунікацій: модуль розроблений для забезпечення обміну трафіком між девайсами та вузлами мережі за допомогою найпопулярніших технологій. Модуль також включає в себе логіку вибору потрібної технології на основі встановлених критерій;
- збору метрик: забезпечує збір інформації про стан девайсів та вузлів, що обєднані в одну мережу, а також її структуризації та збереження.
- конфігурацій: розроблений для забезпечення автоматичної конфігурації та переконфігурації системи;
- керування системою: модуль відповідає за можливість керування системою, обєднує декілька модулів в один блок, та використовує їх можливості;
- роботи з базою даних: реалізовує логіку маніпуляцій з базою даних;
- роботи з командним рядком: модуль надає можливість адміністратору за допомогою командного рядку, впливати на конфігураційні файли та налаштування.

На нижньому шарі знаходяться самі IoT девайси, та реалізація низькорівненового керування ними.

4.2 Вибір бекенд технологій

Реалізація розглянутих модулів потребує певного стеку технології. Було проаналізовано то обрано стек технологій для кожного з модулів, що мають відношення до бекенду системи. Нижче розглянемо обрані технології для кожного з модулів:

– кібербезпека: модуль складається з python скриптів , що взаємодіють з операційною системою. Мова програмування python підходить нам для того, щоб описати механізм взаємодії між модулями системи та сторонніми утилітами, бібліотеками. Міжпроцесорна взаємодія була реалізована на основі сокетів, за допомогою фреймворку Apache Thrift;

– комунікацій : модуль складається з python оболочки над c++ кодом, так як більшість драйверів для девайсів написана на мові програмування C та C++. Міжпроцесорна взаємодія описана на основі фреймворку Apache Thrift та стандартними методами Unix ipc;

– збору метрик: модуль розроблений на основі того ж стеку технологій, що і модуль комунікацій, також використано стандартну бібліотеку C++ та бібліотеку boots.asio;

– конфігурацій : модуль конфігурацій було описано за допомогою мови програмування Python та Bash. Також було використані стандартні Linux утиліти по типу cron, для журналу задач;

– роботи з базою даних: модуль роботи з базами даних було розроблено за допомогою мов програмування C++ та SQL, на основі postgresql;

– роботи з командним рядком: модуль роботи з командним рядком був реалізований за допомогою мови програмування C++, а за допомогою мови програмування Python було розроблено оболочку для легшої роботи з даними та утилітами в середовищі Unix.

4.3 Вибір фронтенд технологій

Розробка користувацького інтерфейсу у вигляді додатку для телефону потребує технології, що дозволять швидко та легко створити додаток, що повністю відповідатиме поставленив вимогам, а саме зрозумілий інтерфейс та візуалізація процесу. Для цих вимог було вибрано мову програмування Python, тому що вибрана мова програмування практикує динамічне типування, а також вона проста у використанні та написанні не великих додатків.

Мобільна розробка на Python поступово прогресує. Результатом цього прогресу є кілька сучасних інструментів, що дозволяють пободувати нативний користувацький інтерфес. Розглянемо деякі фреймворки, що вирішують поставлені задачі:

– Kivy - бібліотека Python з відкритим кодом для розробки міжплатформних програм GUI. Він дозволяє писати графічні програми pure-Python, які працюють на основних платформах настільних ПК (Windows, Linux та macOS) та на iOS та Android. Kivy постачається із створеним на замовлення інструментарієм інтерфейсу користувача, який надає власні версії кнопок, текстових міток, форм введення тексту тощо.

– BeeWare - пропонує набір інструментів та шар абстракції, який можна використовувати для написання власних мобільних та настільних програм за допомогою Python. За допомогою програми BeeWare, користувальницький інтерфейс, який використовує додаток, - це кнопки, прапорці та елементи форми, надаються в основі операційної системи.

Ключова відмінність Kivy від BeeWare полягає в тому, що програми BeeWare використовують вбудований інструментарій інтерфейсу інтерфейсу платформи, на якій вони працюють, тоді як додатки Kivy використовують користувацький інструментарій інтерфейсу користувача, який використовує однакові елементи управління на всіх платформах.

4.4 Вибір програмних технологій

4.3.1 Strongswan

StrongSwan - це повне рішення IPsec, що забезпечує шифрування та автентифікацію серверам та клієнтам. Він може використовуватися для захисту зв'язку з віддаленими мережами, так що віддалене підключення буде таким самим, як підключення локально [19].

– шлюз: шлюз зазвичай є вашим брандмауером, але це може бути будь-який хост у вашій мережі;

– часто шлюз також може обслуговувати невелику мережу з DHCP і DNS;

– клієнти з віддаленим доступом / Roadwarrior: Зазвичай дорожні воїни - це ноутбуки та інші мобільні пристрої підключення від віддаленого до вашої мережі за допомогою шлюзу. Віддалені хости / Host-to-Host: Це може бути віддалений веб-сервер або резервна система. Це проілюстровано на зображенні хостом winnetou та будь-яким із шлюзів;

– зазвичай ініціює будь-який з них.Віддалені сайти / сайт-сайт: хости в двох або більше підмережах у різних місцях повинні мати можливість доступу один одного.

Для того, щоб переконатися, що рівномірний, з яким встановлено IKE_SA, є дійсно тим, хто це стверджує, він повинен бути автентифікований.StrongSwan надає кілька методів для цього:

– Аутентифікація відкритого ключа: для використання автентичності RSA, ECDSA або EdDSA X.509 перевіряється справжність однорангового. Сертифікати можуть бути самостійно підписані (у такому випадку вони повинні бути встановлені на всіх колег) або підписані загальним Орган з сертифікації (CA). Останнє значно спрощує розгортання та конфігурацію, як лише шлюз потрібен сертифікат CA для автентифікації всіх однолітків, які надають дійсний сертифікат, підписаний цим CA. Списки відкликання сертифікатів (CRL) або Інтернет-протокол про статус сертифіката (OCSP) можуть бути використовується для перевірки дійсності

сертифікатів. Для надійного зберігання приватних ключів смарт-карти можуть використовуватися через плагін PKCS # 11. З метою запобігання нападів "посередника" особу, яку стверджує одноліток, слід підтвердити сертифікат або за темою, або за розширенням SubAltName [20].

– Pre-Shared-Key (PSK): Загальнодоступний ключ - це проста можливість розгортання, але для його захисту потрібні чіткі секрети. Якщо PSK відомий багатьом користувачам (що часто буває з IKEv1 XAuth з PSK), будь-який користувач, який знає секрет міг уособлювати ворота. Тому цей метод не рекомендується застосовувати у великих масштабах розгортання. Розширюваний протокол аутентифікації (EAP): Це охоплює декілька можливих методів аутентифікації, деякі є на основі автентифікації користувача / пароля (EAP-MD5, EAP-MSCHAPv2, EAP-GTC) або на сертифікатах (EAP-TLS), деякі навіть можуть тунелювати інші методи EAP (EAP-TTLS, EAP-PEAP).

– Фактична аутентифікація користувачів може бути делегована на сервер RADIUS за допомогою плагіна eap-radius. Автентифікацію EAP можна використовувати лише з IKEv2, а для деяких методів з IKEv1 за допомогою плагіна xauth-eap.

– eXtended Authentication (XAuth): XAuth забезпечує гнучку рамку аутентифікації в IKEv1. Це в основному використовується для автентифікації на основі імені користувача / пароля. Крім того, він зазвичай використовується як другий метод аутентифікації після взаємної автентифікації або з сертифікатами, або з PSK. Однак з IKEv1 гібридна автентифікація є, можливо автентифікувати шлюз за допомогою сертифіката та використовувати тільки XAuth для аутентифікації клієнта. За допомогою IKEv2 можна використовувати декілька раундів аутентифікації, наприклад, спочатку аутентифікувати "машини" за допомогою сертифікати, а потім "користувач" зі схемою автентифікації на основі імені користувача / пароля (наприклад, EAP-MSCHAPv2). Це також можливо використовувати асиметричну аутентифікацію, наприклад, шляхом автентифікації шлюзу з сертифікатом і клієнтом методом EAP на основі імені користувача / пароля (у першому раунді аутентифікації).

Зверніть увагу, що не всі IKEv2 реалізації підтримують це розширення (RFC 4739) [21].

4.3.2 Wpa-supplicant

Бездротові мережі не вимагають фізичного доступу до мережевого обладнання так само, як і дротові мережі. Це полегшує стороннім користувачам пасивний моніторинг мережі та захоплення всіх переданих кадрів. Крім того, несанкціоноване використання мережі набагато простіше. У багатьох випадках це може статися навіть без явного знання користувача, оскільки адаптер бездротової локальної мережі, можливо, був налаштований для автоматичного приєднання до будь-якої доступної мережі.

Шифрування шару зв'язку може використовуватися для забезпечення рівня безпеки бездротових мереж. Оригінальний стандарт бездротової локальної мережі, IEEE 802.11, включав простий механізм шифрування, WEP. Однак це виявилося недоліком у багатьох областях, а мережа, захищена WEP, не може вважатися безпечною. IEEE 802.1X аутентифікація та часто змінені динамічні ключі WEP можуть використовуватися для покращення безпеки мережі, але навіть це успадкувало проблеми безпеки завдяки використанню WEP для шифрування. Захищений доступ Wi-Fi та поправка IEEE 802.11i до стандарту бездротової локальної мережі вводять механізм вдосконалення безпеки бездротових мереж. Мережі з підтримкою IEEE 802.11i, які використовують CCMP (механізм шифрування, заснований на сильному криптографічному алгоритмі AES), нарешті можна назвати захищеним, використовуваним для програм, які потребують ефективного захисту від несанкціонованого доступу [22].

wpa_supplicant - це реалізація компонента WPA Supplicant, тобто частини, яка працює на клієнтських станціях. Він реалізує узгодження ключів WPA з аутентифікатором WPA та автентифікацією EAP з сервером аутентифікації. Крім того, він управляє роумінгом та аутентифікацією / асоціацією IEEE 802.11 драйвера бездротової локальної мережі.

wpa_supplicant розроблений як програма "демон", яка працює у фоновому режимі і виконує функцію резервного компонента, що контролює бездротове з'єднання. wpa_supplicant підтримує окремі програми фронтену, а приклад тексту, заснований на тексті, wpa_cli, включений до wpa_supplicant. Перш ніж wpa_supplicant може виконати свою роботу, мережевий інтерфейс повинен бути доступним. Це означає, що фізичний пристрій повинен бути присутнім та увімкнутим, а драйвер для пристрою повинен бути завантажений. Демон негайно вийде, якщо пристрій ще не доступний. Після того, як wpa_supplicant налаштував мережевий пристрій, може розпочатися конфігурація вищого рівня, наприклад DHCP. Існує безліч способів інтеграції wpa_supplicant в мережні сценарії машини, деякі з яких описані в розділах нижче [23].

Наступні кроки використовуються під час асоціації з AP за допомогою WPA:

- wpa_supplicant просить драйвер ядра сканувати сусідні BSS
- wpa_supplicant вибирає BSS на основі його конфігурації
- wpa_supplicant просить драйвер ядра асоціюватися з обраним BSS
- Якщо WPA-EAP: інтегрований IEEE 802.1X Supplicant завершує автентифікацію EAP з сервером аутентифікації (проксі-аутентифікатор в AP)
- Якщо WPA-EAP: головний ключ отримано від Податника IEEE 802.1X
- Якщо WPA-PSK: wpa_supplicant використовує PSK як основний ключ сеансу
- wpa_supplicant завершує рукостискання з 4-х сторін WPA і груповий клавіш WPA аутентифікатором (AP)
- wpa_supplicant налаштовує ключі шифрування для одноадресної передачі та трансляції
- звичайні пакети даних можуть передаватися та прийматись

4.3.2 Freeradius

FreeRADIUS - найпопулярніший і найпоширеніший сервер RADIUS у світі. Він служить основою для безлічі комерційних пропозицій, і він забезпечує потреби в

аутентифікації, авторизації та обліку (AAA) багатьох компаній Fortune 500 та провайдерів першого рівня. Він також широко використовується академічним співтовариством (наприклад, Eduroam, всесвітня послуга доступу до роумінгу, розроблена для міжнародного науково-освітнього співтовариства, використовує програмне забезпечення FreeRADIUS). FreeRADIUS був розроблений з використанням мдульної конструкції для заохочення активнішої участі громади [24].

Популярність FreeRADIUS можна віднести до безлічі додаткових переваг, які він пропонує, набагато вище, ніж за тими, що знайдені у широкому спектрі інших серверів RADIUS. FreeRADIUS заснований на багатофункціональному, модульному та масштабованому протоколі дизайну, який надає наступні переваги та переваги мережним адміністраторам:

– FreeRADIUS підтримує більше типів аутентифікації, ніж будь-який інший сервер з відкритим кодом. Наприклад, FreeRADIUS - єдиний сервер RADIUS з відкритим кодом, що підтримує протокол розширюваної автентифікації (EAP). FreeRADIUS також є єдиним сервером RADIUS (комерційним або відкритим кодом), який підтримує "віртуальні сервери". Використання віртуальних серверів означає, що складні реалізації спрощуються, а постійні витрати на підтримку та обслуговування для адміністраторів мережі значно скорочуються; таким чином, можливість FreeRADIUS підтримувати віртуальні сервери надає йому величезну перевагу перед конкуренцією.

– Модульна конструкція робить FreeRADIUS легким для розуміння. Модульний інтерфейс також спрощує додавання або видалення модулів. Наприклад, якщо функція не потрібна для певної конфігурації, модуль легко видаляється. Після видалення модуля це не впливає на продуктивність сервера, використання пам'яті або безпеку. Ця гнучкість дозволяє серверу працювати на платформах, починаючи від вбудованих систем до багатоядерних машин з гігабайтами оперативної пам’яті.

– Один сервер RADIUS може легко перейти від обробки одного запиту кожні кілька секунд до обробки тисяч запитів в секунду, просто переконфігурувавши кілька налаштувань за замовчуванням. Багато великих організацій (ті, у кого більше 10 мільйонів клієнтів) залежать від FreeRADIUS для своїх потреб AAA. Хоча багато

комерційних постачальників пропонують різні версії свого програмного забезпечення для задоволення різних потреб, для отримання кращої продуктивності, більшої кількості областей, більше клієнтів RADIUS та багатьох інших функцій потрібна лише остання версія FreeRADIUS, без необхідності купувати додаткові ліцензії на продукт.

4.4 Вибір сховищ даних

Існує два основні типи баз даних, реляційні та нереляційні. Основна різниця між ними полягає в тому, як вони обробляють дані.

Реляційні бази даних структуровані. У вас є таблиці, і ці таблиці можуть мати залежність один від одного або відносини. У базі даних магазину буде таблиця для клієнтів та одна для замовлень. Ці дві таблиці пов'язані між собою, оскільки замовлення робить замовник.

Нереляційні бази даних орієнтовані на документи. Це так зване сховище типу документа дозволяє зберігати декілька "категорій" даних в одній конструкції чи документі. Отже, використовуючи попередній приклад, документ Замовника, містила б інформацію про замовника, підкатегорію для всіх їх замовлень тощо.

4.4.1 Реляційні бази даних SQL

Реляційна база даних використовує SQL, що є коротким для структуризованої мови запитів. Це досить жорсткий і стандартний спосіб зберігання даних за допомогою таблиць, стовпців і рядків. Стовпці представляють точку даних, яку потрібно зберігати, а рядок являє собою запис з точками даних на стовпець, який був визначений. Вони визначені в таблиці, яка має атомний характер; це означає, що таблиця дійсно повинна зберігати записи даних лише про одну сутність або об'єкт одночасно. Коли потрібні додаткові деталі або дані повинні бути пов’язані із записом з однієї таблиці в іншу, тоді ми маємо те, що ми називаємо відносинами. Загальний

ключ встановлюється між двома (або більше) таблицями, і він використовується для цього об'єднання там після.

Популярні реляційні системи баз даних SQL:

- Microsoft SQL Server
- Oracle
- MySQL / MariaDB
- PostgreSQL
- Microsoft Azure SQL

Розглянемо випадки коли реляційні бази даних SQL кращі ніж не реляційні:

– використовуються принципи ACID (Atomicity, Consistency, Isolation, Durability). Це зменшує аномалії, застосовує цілісність, і саме тому це є кращим для комерційних та фінансових програм.

– Структура даних не змінюється. Дизайн додатків є надійним і не очікується, що він буде змінюватися з майбутніми вимогами (принаймні, не дуже часто).

4.4.2 Нереляційні бази даних NoSQL

На відміну від реляційних аналогів, бази даних без sql дозволяють набагато більше гнучкості та адаптивності при розробці програми. Якщо ваші вимоги до даних не зрозумілі з самого початку або ви маєте справу з величезною кількістю неструктурованих даних, можливо, у вас не буде розкоші розробити реляційну базу даних з чітко визначеними таблицями та взаємозв'язками. Бази даних NoSQL орієнтовані на документи. Замість використання таблиць ці документи дозволяють нам зберігати неструктуровані дані в одному документі. Таким чином, документ може містити реквізити замовника, а також усі їхні замовлення на сьогоднішній день, їхні улюблені тощо. Thi більш інтуїтивно зрозумілий і вимагає менше перестрибування через таблиці, щоб знайти всі дані, що стосуються клієнта. Однак зауважте, що це призводить до необхідності додаткових зусиль для обробки та

збільшення обсягу зберігання у міру збільшення розмірів документів. Зберігання не буде настільки організованим у реляційній базі даних.

Популярні нереляційні бази даних NoSQL:

- MongoDB
- Oracle NoSQL
- Apache CouchDB
- Redis

Розглянемо випадки коли не реляційні бази даних NoSQL кращі ніж реляційні:

– швидка розробка додатків. База даних NoSQL, що підтримує швидко змінюються дизайни та кодування спринтів, і ідеально підходить для більш Agile налаштувань, де вимоги часто змінюються;

– велика кількість даних з малою структурою. Так само, як висловлено в попередньому пункті, якщо вимоги до даних не зрозумілі, але потрібно десь і якось зберігати багато даних, можна використовувати цей тип бази даних, який може бути перетвореним на льоту, щоб відповідати вимозі.

4.4.3 Бекенд сховища даних

Зараз найпопулярнішими реляційними СУБД виступають MySQL, PostgreSQL та Oracle, але лише MySQL та PostgreSQL є повністню безкоштовними для використання та існуються у вільному достіпу.

Отже, для більшої рентабельності, будемо порівнювати саме ці системи управління баз даних.

Порівняльна характеристика MySQL та PostgreSQL розглянута нижче (Таблиця 4.2) [25]..

Таблиця 4.1 – порівняння сховищ даних

Можливісті системи управління базами даних	PostgreSQL	MySQL
Реплікація	Фізична, master-slave	Логічна, master-master та master-slave, виникають проблеми зі стабільністю та швидкодією
Документація	Детальна	Не достатня, виникають задачі, які змушують нас навідуватись за допомогою до сторонніх сайтів
Підтримка стандартів SQL	Підтримує стандарти SQL-92, SQL-98, SQL-2003, а також планується підтримка SQL-2011	Не підтримує жоден з стандартів
Наявність оптимізатора запитів	присутній	відсутній, присутній тільки парсер та нормальнізація запитів
Складність адміністрування	Складне, через велику кількість можливих налаштувань	просте, через не велику кількість можливих налаштувань
Створення індексів неблокуючим шляхом	присутнє	відсутнє
Паралельне планування запитів	присутнє	відсутнє
Часткові індекси	присутні	відсутні
Наслідування таблиць	присутнє	відсутнє
Перевантаження функцій	присутнє	відсутнє

Продовження таблиці 4.1

Можливісті системи управління базами даних	PostgreSQL	MySQL
Підтримувані операційні системи	Linux OS X Solaris Unix Windows NetBSD OpenBSD FreeBSD HP-UX	FreeBSD OS X Solaris Windows Linux
Схема даних	присутня	присутня
Типи даних	підтримуються	підтримуються
Вторинні індекси	присутні	присутні
Підтримка SQL	присутня	присутня
Наявні API	ODBC JDBC Нативна C бібліотека ADO.NET	ADO.NET Своє нативне API JDBC ODBC
Підтримувані мови програмування	.NET C C++ Delphi Java info JavaScript (Node.js) Perl PHP Python Tcl	Ada C C# C++ Erlang Haskell Java JavaScript (Node.js) Objective-C OCaml Perl PHP Python Ruby

Продовження таблиці 4.1

Можливісті системи управління базами даних	PostgreSQL	MySQL
Розділення даних	розділення за допомогою діапазонів,списків, хешів	горизонтальне розділення та шардування
Мова написання	C	C та C++

З таблиці зрозуміло, що PostgreSQL має значні переваги перед MySQL, тому в реалізації системи буде використано саме PostgreSQL

4.4.4 Фронтенд сховища даних

Також потрібно вибрати не реляційну базу даних, для швидкого розроблення тестового додатку для користувача.

Порівняльна характеристика MongoDB та Redis розглянута нижче (Таблиця 4.2)[26].

Таблиця 4.2 – порівняння сховищ даних

Можливісті системи управління базами даних	MongoDB	Redis
Реплікація	master-slave	master-slave, Multi-master
Документація	детальна й повноцінна	детальна й повноцінна
Типи даних	підтримуються	частково
Підтримка SQL	SQL-запити лише для читання через з'єднувач MongoDB для BI	не підтримує

Мова написання	C++	C
Наявні API	власний протокол з використанням JSON	власний протокол
Вторинні індекси	наявні	наявні
Підтримувані операційні системи	Linux OS X Solaris Windows	BSD Linux OS X Windows
Підтримувані мови програмування	Actionscript C C# C++ ColdFusion D Go Groovy Haskell Java JavaScript MatLab Perl PHP PowerShell Python R Ruby	C C# C++ Go Haskell Java JavaScript (Node.js) MatLab Objective-C Pascal Perl PHP Prolog Python R Ruby Rust Scala Swift

	Scala Smalltalk	
Паралельне планування запитів	наявне	наявне

З таблиці хоч зразу і не зрозуміло, котра з СУБД краще буде для використання, але якшо оглянути детальніше то можна помітити, що MongoDB надає більше можливостей, а також дуже важливими факторами є те, що MongoDB підтримує типи даних та має власний протокол з використанням JSON.

5. РОЗРОБКА СИСТЕМИ

5.1 Сценарії використання системи

У даному підрозділі розглянуті сценарії використання системи та розроблено діаграму сценаріїв системи. Діаграму сценаріїв із функціональними вимогами до системи наведено у додатку [], у таблицях нижче наведено опис прецедентів (таблиці 5.1 - []).

Таблиця 5.1 - Сценарій використання

Назва	Перегляд профілю користувача
ID	1
Опис	Користувач потрапляє на екран з інформацію про профіль користувача після логіну в систему через додаток
Актори	Користувач
Вигоди компанії	Якщо користувач не знатиме стан системи та не зможе налаштувати систему то він не зможе її використовувати
Частота користування	Постійно, користувач потрапляє на екран власного профілю зразу після логіну
Тригери	-
Передумови	Користувач проходить крок логіну
Постумови	-
Основний розвиток	Користувач потрапляє в головне меню додатку

Винятки	-

Таблиця 5.2 - Сценарій використання

Назва	Перегляд інформації про конкретний хост
ID	2
Опис	Після натискання на обраний користувачем хост, користувач отримує доступ до детальної інформації про хост
Актори	Користувач
Вигоди компанії	Інформація про ip та mac адреси , а також стан даного хосту в реальному часі буде корисний для моніторингу системи.
Частота користування	Періодично
Тригери	Користувач два рази натискає на потрібний йому хост з списку хостів
Передумови	Користувач успішно залогінився в систему
Постумови	-
Основний розвиток	Користувач отримує інформацію про потрібний йому хост
Альтернативні розвитку	-

Винятки	-

Таблиця 5.3 - Сценарій використання

Назва	Підключення нового хосту в підмережу
ID	3
Опис	Користувач виконує додавання нового девайсу зі списку доступних йому девайсів
Актори	Користувач
Вигоди компанії	Без підключення нових девайсів не можливе масштабування підмережі та розширення потрібного користувачу функціоналу
Частота користування	Періодично
Тригери	Користувач натискує клавішу додати девайс
Передумови	Користувач знаходиться в головному меню
Постумови	В головному меню зявляється новий елемент в списку підключених девайсів

Основний розвиток	Користувач додає новий девайс в мережу з списку доступних йому девайсів
Альтернативні розвитку	Адміністратор додасть новий девайс до мережі
Винятки	Адміністратор вимкнув можливість додавати новий девайс в мережу для користуваців без прав адміністратору

Таблиця 5.4 - Сценарій використання

Назва	Моніторинг системи
ID	4
Опис	Користувач потрапляє на екран моніторингу системи, де відображається інформація про девайси
Актори	Користувач
Вигоди компанії	Частина основного мінімальному функціоналу, що надає користувачу можливість оцінити правильність роботи мережі
Частота користування	Періодично
Тригери	Користувач натискає на елемент головного меню
Передумови	Користувач знаходиться в головному меню
Постумови	Змінюється екран, користувача переносить на екран моніторингу
Основний розвиток	Користувач отримує можливість подивиться, що відбувається з його мережею в даний момент часу
Альтернативні розвитку	-
Винятки	-

Таблиця 5.5 - Сценарій використання

Назва	Перегляд метрик
ID	5
Опис	Користувач потрапляє на екран перегляду метрик, що надсилаються з кожного девайсу підключеного до мережі
Актори	Користувач
Вигоди компанії	Частина функціоналу по моніторингу системи та аналізу отриманих метрик з девайсів
Частота користування	Рідко
Тригери	Користувач двічі натискає на потрібний йому елемент з списку девайсів
Передумови	Користувач знаходиться на екрані моніторингу системи
Постумови	Отримана можливість продивитись детальну інформацію про стан девайсу в реальному часі
Основний розвиток	Користувач знаходячись на екрані моніторингу системи вибирає потрібний йому девайс, двічі натиснувши елемент зі списку
Альтернативні розвитку	Адміністратор має можливість перегляду метрик з командного рядку

Винятки	Потрібний девайс знаходиться оффлайн

Таблиця 5.6 - Сценарій використання

Назва	Аналіз трафіку девайсу
ID	6
Опис	
Актори	Користувач
Вигоди компанії	Частина основного мінімального функціоналу по моніторингу стану системи
Частота користування	Періодично
Тригери	Користувач натискає на елемента екрані з назвою аналіз трафіку
Передумови	Користувач знаходиться на екрані моніторингу системи
Постумови	Користувача отримує результати аналізу трафіку

Основний розвиток	Користувач знаходячись на екрані моніторингу системи обирає потрібний йому девайс зі списку девайсів, потім натискає на елемент під назвою аналіз трафіку девайсу
Альтернативні розвитку	-
Винятки	Девайс знаходиться оффлайн

Таблиця 5.7 - Сценарій використання

Назва	Керування системою
ID	7
Опис	Користувач переходить на екран керування системою
Актори	Користувач
Вигоди компанії	Частина мінімального функціоналу по керуванню системою
Частота користування	Періодично

Тригери	Користувач натискає на елемент головного меню
Передумови	Користувач знаходиться в головному меню
Постумови	Змінюється екран, користувача переносить на екран керування системою
Основний розвиток	Користувач знаходячись на головному екрані додатку обирає елемент головного меню за назвою керування системою
Альтернативні розвитку	-
Винятки	Адміністратор вимкнув можливість керування системою для користувача

Таблиця 5.8 - Сценарій використання

Назва	Налаштування девайсу
ID	8
Опис	Користувач змінює налаштування девайсу
Актори	Користувач
Вигоди компанії	Частина функціоналу керуванню системою
Частота користування	Рідко

Тригери	Користувач двічі клікає на потрібний елемент зі списку девайсів
Передумови	Користувач знаходиться на екрані керування системою
Постумови	Можливість змінити налашування конкретного девайсу
Основний розвиток	Користувач знаходячись на екрані керування системою двічі клікає на потрібний елемент зі списку девайсів
Альтернативні розвитку	Адміністратор змінює налаштування девайсу з командного рядку
Винятки	Адміністратор заборонив змінювати налаштування девайсу

Таблиця 5.9 - Сценарій використання

Назва	Переконфігурація системи
ID	9
Опис	Адміністратор викликає процес автоматичної переконфігурації системи
Актори	Адміністратор
Вигоди компанії	Частина функціоналу по керуванню системою

Частота користування	Рідко
Тригери	Адміністратор вводить команду переконфігурації системи в командний рядок
Передумови	Користувач зайшов в систему з правами адміністратора через командний рядок
Постумови	Переконфігурована система
Основний розвиток	Адміністратор знаходячись в командному рядку вводить команду для переконфігурації системи
Альтернативні розвитку	-
Винятки	В системі не зарегістровано жодного девайсу

Таблиця 5.10 - Сценарій використання

Назва	Додавання сертифікатів
ID	10
Опис	Адміністратор може додати новий сертифікат для авторизації

Актори	Адміністратор
Вигоди компанії	Частина основного мінімальному функціоналу
Частота користування	Рідко
Тригери	Адміністратор вибирає сертифікат з списку доступних сертифікатів
Передумови	Користувач зайшов в систему з правами адміністратора через командний рядок
Постумови	Появлення в конфігурації нового сертифікату
Основний розвиток	Адміністратор знаходячись в командному рядку вводить команду додавання сертифікатів
Альтернативні розвитку	-
Винятки	Адміністратор вибрав не захищений режим роботи системи

Таблиця 5.11 - Сценарій використання

Назва	Видалення сертифікату
ID	11

Опис	Адміністратор може видалити сертифікат з списку сертифікатів
Актори	Адміністратор
Вигоди компанії	Частина основного мінімальному функціоналу
Частота користування	Рідко
Тригери	Адміністратор вводить команду для видалення сертифікату із системи через командний рядок
Передумови	Користувач зайшов в систему з правами адміністратора через командний рядок
Постумови	Видалення елементу з списку сертифікатів
Основний розвиток	Адміністратор знаходячись в командному рядку вводить команду видалення сертифікату, після чого отримує список існуючих сертифікатів та обирає сертифікат який потрібно видалити
Альтернативні розвитку	-
Винятки	-

Таблиця 5.12 - Сценарій використання

Назва	Генерація нового сертифікату
ID	12
Опис	Адміністратор може згенерувати сертифікат
Актори	Адміністратор
Вигоди компанії	Частина основного мінімального функціоналу
Частота користування	Рідко
Тригери	Адміністратор вводить команду генерації нового сертифікату в командному рядку
Передумови	Користувач зайшов в систему з правами адміністратора
Постумови	Створення нового сертифікату
Основний розвиток	Адміністратор знаходиться в командному рядку, вводить команду генерації нового сертифікату, після чого обирає алгоритм шифрування із списку алгоритмів доступних в системі
Альтернативні розвитку	-
Винятки	-

Таблиця 5.13 - Сценарій використання

Назва	Генерація нового приватного ключа
ID	13
Опис	Адміністратор може генерувати приватний ключ, яким підписується сертифікат
Актори	Адміністратор
Вигоди компанії	Система захисту інформації за допомогою сертифікованої передачі даних, це один з кроків збільшення приватності та захищеності системи
Частота користування	Рідко
Тригери	Адміністратор вводить команду генерації нового приватного ключа в командний рядок
Передумови	Користувач зайшов в систему з правами адміністратора
Постумови	Створення нового приватного ключа
Основний розвиток	Адміністратор знаходиться в командному рядку, вводить команду генерації нового приватного ключа, після чого обирає алгоритм шифрування із списку алгоритмів доступних в системі
Альтернативні розвитку	-

Винятки	-

Таблиця 5.14 - Сценарій використання

Назва	Підпис сертифікату
ID	14
Опис	Адміністратор підписує сертифікати приватним ключем зі списку
Актори	Адміністратор
Вигоди компанії	Частина функціоналу по захисту інформації
Частота користування	Рідко
Тригери	Адміністратор вводить команду підпису сертифікату
Передумови	Користувач зайшов в систему з правами адміністратора
Постумови	Підписаний сертифікат
Основний розвиток	Адміністратор знаходиться в командному рядку, вводить команду підпису сертифікату, після чого обирає приватний ключ, яким буде підписано сертифікат із списку доступних ключів
Альтернативні	-

розвитку	
Винятки	-

Таблиця 5.15 - Сценарій використання

Назва	Збір метрик
ID	15
Опис	Адміністратор включає механізм збору метрик та збереження в базу даних
Актори	Адміністратор
Вигоди компанії	Частина функціоналу по захисту інформації
Частота користування	Рідко
Тригери	Адміністратор вводить команду запуску збору метрик в командному рядку
Передумови	Користувач зайшов в систему з правами адміністратора
Постумови	Всі метрики, що потрапляють на головний сервер будуть зберігатись в базу даних для подальшого опрацювання користувачем

Основний розвиток	Адміністратор знаходиться в командному рядку, вводить команду для початку збору метрик
Альтернативні розвитку	-
Винятки	-

Таблиця 5.16 - Сценарій використання

Назва	Зміна DHCP IP
ID	16
Опис	Адміністратор змінює IP адрес DHCP серверу
Актори	Адміністратор
Вигоди компанії	Частина функціоналу по конфігурації мережі
Частота користування	Рідко
Тригери	Адміністратор вводить команду зміни IP адреси DHCP серверу
Передумови	Користувач зайшов в систему з правами адміністратора
Постумови	Змінюється адрес DHCP серверу, після чого система починає використовувати новий сервер

Основний розвиток	Адміністратор знаходиться в командному рядку, вводить команду для зміни IP адреси DHCP серверу
Альтернативні розвитку	-
Винятки	-

Таблиця 5.17 - Сценарій використання

Назва	Вибір комунікаційного протоколу за замовчуванням
ID	17
Опис	Адміністратор обирає протокол комунікації за замовчуванням
Актори	Адміністратор
Вигоди компанії	Частина функціоналу по захисту інформації
Частота користування	Рідко
Тригери	Адміністратор вводить команду встановлення протоколу комунікації за замовчуванням

Передумови	Користувач зайшов в систему з правами адміністратора
Постумови	Новий протокол комунікації використовуєть за замовчуванням
Основний розвиток	Адміністратор знаходиться в командному рядку, вводить команду встановлення нового протоколу комунікації за замовчуванням
Альтернативні розвитку	-
Винятки	-

Таблиця 5.18 - Сценарій використання

Назва	Вибір стандарту захисту
ID	18
Опис	Адміністратор може обрати стандарт захисту
Актори	Адміністратор
Вигоди компанії	Частина основного мінімальному функціоналу
Частота користування	Рідко

Тригери	Адміністратор вводить команду вибору стандарту захисту
Передумови	Користувач зайшов в систему з правами адміністратора
Постумови	Система починає використовувати обраний стандарт захисту
Основний розвиток	Адміністратор вводить команду для вибору стандарту захисту мережі та приватної інформації, після чого обирає стандарт з списку доступних
Альтернативні розвитку	-
Винятки	-

5.2 Розробка баз даних

Для реалізації системи було обрано два види баз даних:

1) реляційна база даних потрібна для збереження налаштувань, конфігруацій та інформації користувача. Також реляційна база даних була використана для зберігання різних метрик, що надсилаються від девайсів до головного серверу. Модель бази даних описано в наступному розділі;
2) нереляційна mongodb база даних використана для зберігання інформації користувача в додатку на телефоні;

5.3 Опис концептуальної моделі даних

Проектування баз даних, процес що складається з декількох етапів, правильне виконання яких забезпечує оптимальнета ефективне використання ресурсів при дальнішому використанні бази даних.

Для того, щоб краще дослідити предметну область було побудовано ER-діаграму перша частина якої представлена на рисунку 5.1. ER-діаграма дозволяє описувати концептуальні схеми за допомогою узагальнених конструкцій блоків, виконує функцію аналізу предметної області, а також початкового етапу проектування бази даних, в якому визначені найголовніші сутності, над якими будуть визначитися атрибути та зв’язки.

База даних була спроектована за допомогою методу «сутність – зв’язок». Була проаналізована предметна область та виділено наступні сутності:

- device
- user
- network
- settings
- certificate
- profile
- metrics

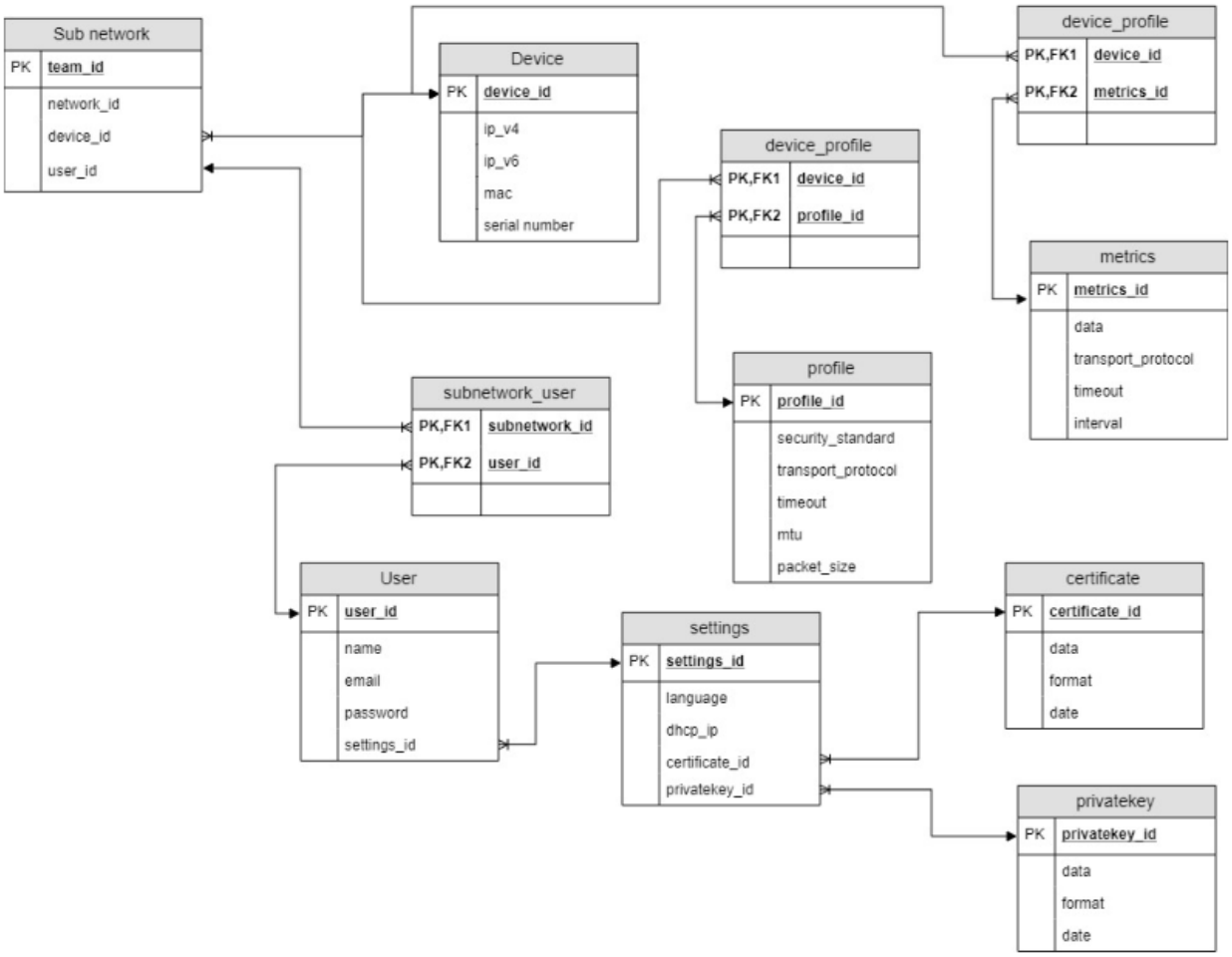

Рисунок 5.1 – ER-діаграма

Наступна частина ER-діаграми відповідає за метрики, що описані в системі , вона представлена на рисунку 5.2. Для неї також була проаналізована предметна область та виділено наступні сутності:

- device
- info
- sensors
- states
- network config
- timers
- device load

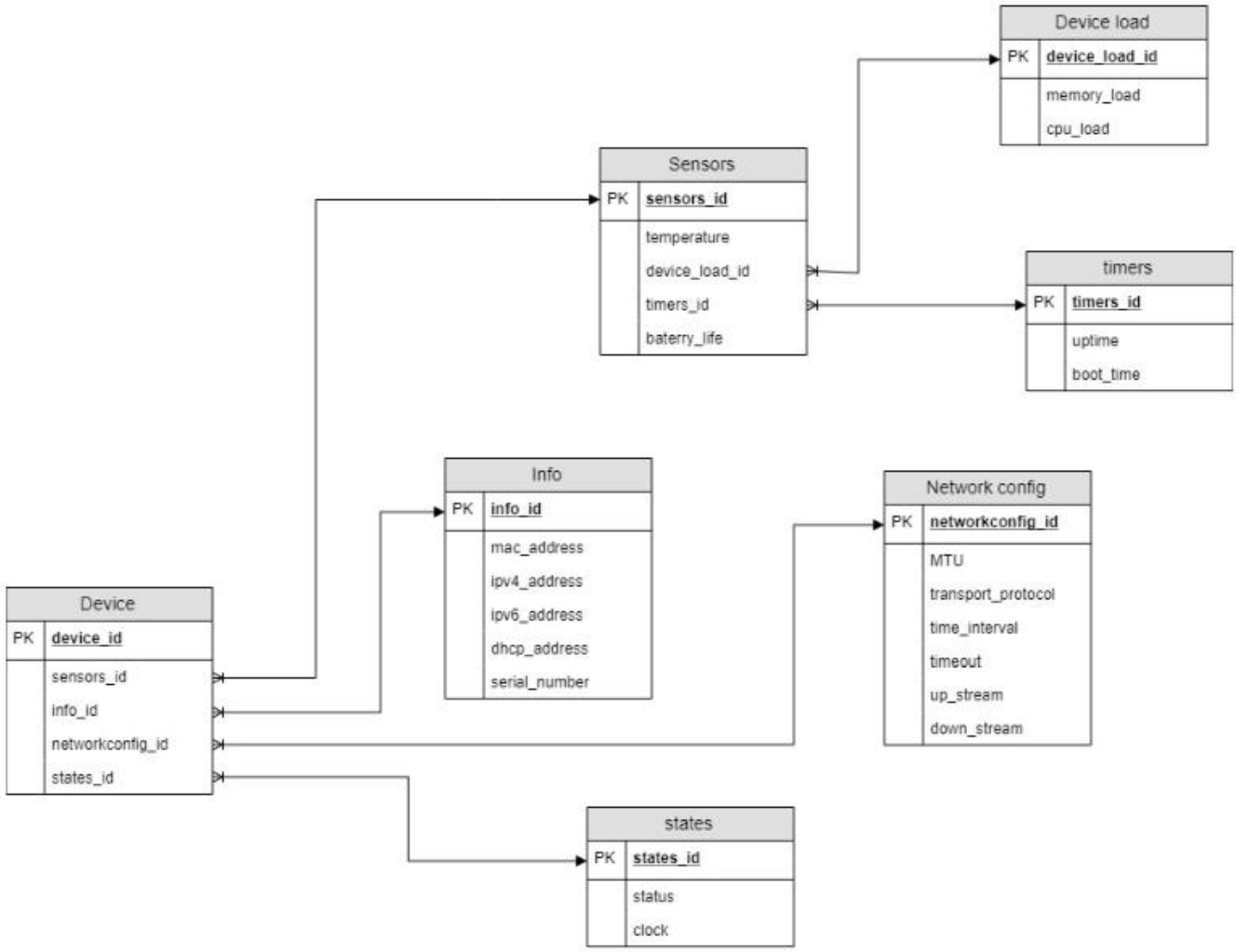

Рисунок 5.2 – ER-діаграма

5.4 Розробка демону автоматичного конфігурування системи

Система працюватиме справно тільки у випадку, якшо налаштування та конфігурації всії програм та технологій будуть правильно налагоджені. Користувач не повинен відповідати за конфігурацію та налаштування, система повинна розпочати роботу навіть без первинного втручання користувача. Саме для вирішення цієї проблеми було розроблено можливість автоматичного конфігурування системи.

Основні сторонні програми які потрібно налаштувати перед запуском системи та при додаванні нового девайсу чи хосту:

– freeradius: конфігурація для RADIUS серверу потрібна для забезпечення автентифікації;

– dhclient: конфігурується для забезпечення автоматичного розподілення IP адресів, що необхідно для роботи в мережі TCP/IP;

– wpa_supplicant: конфігурується для забезпечення узгодження ключів шифрування з аутентифікатором. Працює в парі з FreeRadius сервером.

5.4.1 Налаштування Dhclient

Автоматичне конфігурування dhclient було досягнуто за допомогою запису декількох видів конфігураційних файлів, це значить, що система не буде генерувати зміст файлу самостійно з самого нуля, існуватиме список з профілів, які система зможе обирати на основі того, для якого з елементів системи конфігурація потребується в даний момент часу. На рисунку 5.3 зображено один з списку профілів, які описані в системі.

```
option rfc3442-classless-static-routes code 121 = array of unsigned integer 8;

send host-name = gethostname();
request subnet-mask, broadcast-address, time-offset, routers,
        domain-name, domain-name-servers, domain-search, host-name,
        dhcp6.name-servers, dhcp6.domain-search, dhcp6.fqdn, dhcp6.sntp-servers,
        netbios-name-servers, netbios-scope, interface-mtu,
        rfc3442-classless-static-routes, ntp-servers;

#send dhcp-client-identifier 1:0:a0:24:ab:fb:9c;
#send dhcp-lease-time 3600;
#supersede domain-name "fugue.com home.vix.com";
#prepend domain-name-servers 127.0.0.1;
#require subnet-mask, domain-name-servers;
timeout 300;
#retry 60;
#reboot 10;
#select-timeout 5;
#initial-interval 2;
#script "/sbin/dhclient-script";
#media "-link0 -link1 -link2", "link0 link1";
#reject 192.33.137.209;
```

Рисунок 5.3. – Конфігурація dhclient.

5.4.2 Налаштування wpa_supplicant

Для налаштування wpa_supplicant демону було розроблено модуль кібербезпеки на мові програмування python. Стартову конфігурацію демону зображено на рисунку 5.4 . В статову конфігурацію входять лише ідентифікатор девайсу, а також шляхи до приватних ключів та сертифікатів, що були згенеровані раніше.

```
network={
        identity="myloginname"
        ca_cert="/etc/cert/CA_cert_chained.pem"
        client_cert="/etc/cert/pebble_device.pem"
        private_key="/etc/cert/pebble_device.prv.pem"
        phase1="tls_disable_time_checks=1"
}
```

Рисунок 5.4 – Конфігурація wpa_supplicant.

На рисунку 5.5 видно функцію, котра задає початкові параметри зміних. Функція використовує командний інтерфейс wpa_cli, за допомогою якого виставляє конфігурацію, починаючи від таймерів, та закінчуючи видом захисту мережі.

```
def configure_wpa_supplicant(self,failure):
    eapol_max_start = 3     # EAPOL::maxStart - Max number of retry iterations
    eapol_start_period = 10 # EAPOL::startPeriod - Timer timeout (sec)
    check_wpa_supp_conn = "wpa_cli ping"

    set_wpa_cli_eapol_vars = "wpa_cli SET EAPOL::maxStart {};" \
                             "wpa_cli SET EAPOL::startPeriod {};" \
                             "wpa_cli set_network 0 key_mgmt IEEE8021X;" \
                             "wpa_cli set_network 0 eap TLS;" \
                             "wpa_cli set_network 0 eapol_flags 0;" \
                             "wpa_cli enable_network 0;" \
                             "wpa_cli select_network 0".format(eapol_max_start, eapol_start_period)

    try:
        subprocess.check_call(check_wpa_supp_conn, shell=True, stdout=subprocess.PIPE)

    except subprocess.CalledProcessError:
        failure = True
        # Timeout for 2 sec if wpa_cli can't ping wpa_supplicant
        time.sleep(1)
    else:
        subprocess.Popen(set_wpa_cli_eapol_vars, shell=True, stdout=subprocess.PIPE,stderr=subprocess.PIPE)
        failure = False

    return failure
```

Рисунок 5.5 – Функція задавання конфігурації для wpa_supplicant демону .

5.4.3 Налаштування Freeradius

Для того, щоб реалізувати захист мережі на основі сертифікованої аутентифікації, потрібно налаштувати не тільки клієтську частину, яка в нашому випадку реалізовується за допомогою wpa_supplicant демону, а також і серверну частину. На рисунку 5.6 зображена схема за якою працюватиме наша система, для захисту мережі.

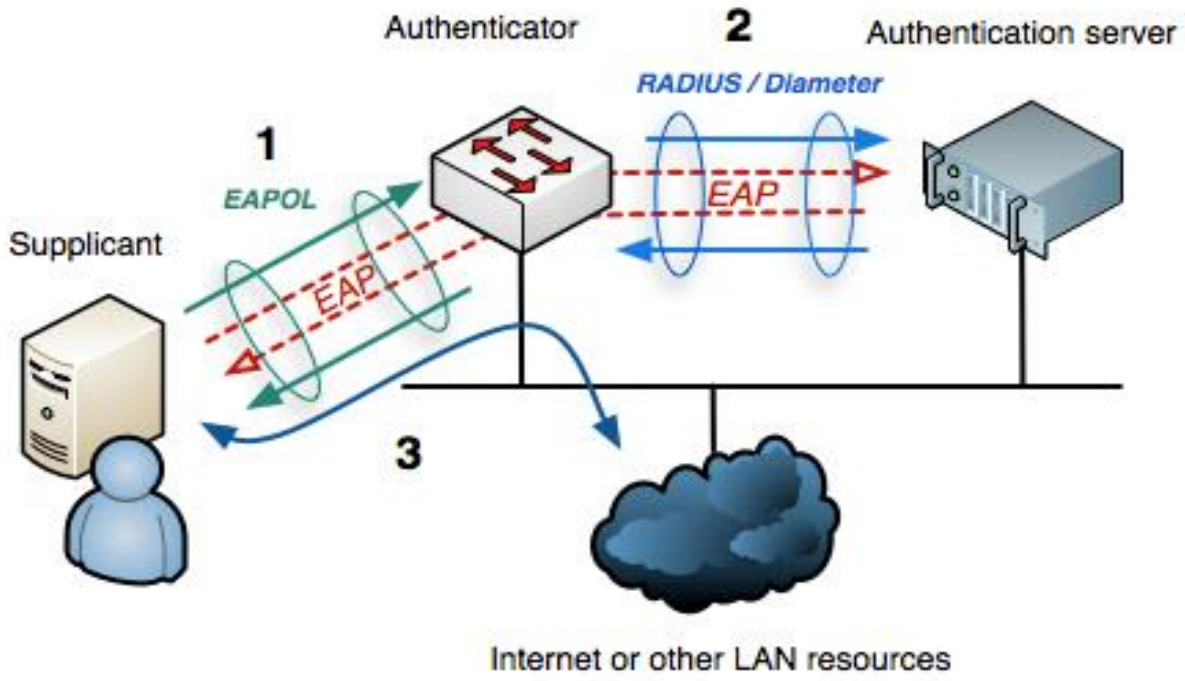

Рисунок 5.6 – Функція задавання конфігурації для wpa_supplicant демону .

На рисунку 5.7 видно мінімальні налаштування для окремого девайсу, які повинен внести адміністратор для правильноної роботи системи.

```
interface Ethernet
   description Dvc 863800 | mue-adle-rpd-dca-7
   load-interval 30
   mtu 2200
   flowcontrol receive on
   switchport access vlan 2005
   mac access-group RPD in
   qos trust dscp
   spanning-tree portfast
   dot1x pae authenticator
   dot1x reauthentication
   dot1x port-control auto
   dot1x timeout reauth-period 15
   dot1x max-reauth-req 6
!
```

Рисунок 5.7 – Функція задавання конфігурації для wpa_supplicant демону .

5.5 Розробка збору метрик

Маніпуляція метриками – це один з основних вимог до системи, так як без цього керування, аналіз та моніторинг системи стає неможливими. Для реалізації поставленої задачі було розроблено протокол обміну метриками TLV на основі фреймворку Apache Thrift. Назва розшифровується тип, довжина та значення. Протокол працює на транспортному рівні на базі протоколу TCP.

Таблиця 5.18 – Специфікація описаних TLV

Номер tlv	Назва tlv	Опис
1	sensors	Комплексна tlv, що існує для обєднання сенсорів в одну метрику
1.1	temperature	Температура девайсу в даний момент часу

Продовження таблиці 5.18

Номер tlv	Назва tlv	Опис
1.2	device_load	Комплексна tlv, що описує навантаження на девайс
1.2.1	memory_load	Навантаження на оперативну память девайсу, кількість використаних мегабайт памяті на момент часу
1.2.2	cpu_load	Навантаження на процесор девайсу
1.3	timers	Комплексна tlv, що описує таймери
1.3.1	uptime	Таймер, що відображає час який девайс знаходиться в онлайн режимі
1.3.2	boot_time	Таймер, що відображає час який витрачається на запуск девайсу
1.4	baterry_life	Стан заряду батареї
2	info	Комплексна tlv, що обєднує інформацію про девайс
2.1	mac_address	Mac адреса девайсу
2.2	ipv4_address	Ipv4 девайсу
2.3	ipv6_address	Ipv6 девайсу
2.4	dhcp_address	Ip адресса dhcp серверу

Продовження таблиці 5.18

Номер tlv	Назва tlv	Опис
2.5	serial_number	Серійний номер девайсу
3	networks	Комплексна tlv, що існує для обєднання інформацію про налаштування мережі
3.1	mtu	Пропускна здатність
3.2	transport_protocol	Транспортний протокол який вибрано по замовчуванню
3.3	time_interval	Інтервал, який показує на валідність метрики та даних отриманих від девайсів
3.4	timeout	Час між відсиланням запиту на обновлення метрик
3.5	up_stream	Швидкість вигрузки даних
3.6	down_stream	Швидкість загрузки даних
4	states	Комплексна tlv, що описує стан девайсу
4.1	status	Статус девайсу, онлайн чи оффлайн
4.2	clock	Статус девайсу, синхронізований за часом чи ні

Для створення Thrift служби для початку треба написати Thrift файли, які описують його, потім згенерувати код на вихідній мові та вказати команди запуску сервера, викликавши після чого їх в клієнті. Розглянемо цей процес на прикладі однієї з специфікованих TLV з таблиці.

```
struct device_load
{
    1: u16  memory_load;
    2: u16  cpu_load;
}
struct timers
{
    1: u16  boot_time;
    2: u16  uptime;
}
struct sensors
{
    1: u16  temperature;
    2: device_load deviceload;
    3: timers timers_;
    4: optional ubyte baterry_life;
}
```

Рисунок 5.8 – Приклад опису структур tlv.

На рисунку 5.8 зображено приклад описання однієї з метрик за допомогою фреймворку Apache Thrift, як можна помітити змінна battery_life помічена як опціональна, це зроблено для того, щоб виділи девайси, які підключені до постійного джерела живлення.

```
bool RpdSysInfoReqHandler::handleRexImmediately(const gcp::thrift::RequestSequence& sequence, gcp::thrift::Sequence& responseSequence)
{
  gcp::thrift::sensors sensor;
  gcp::thrift::device_load deviceload;
  gcp::thrift::device_load timers;
  if(sequence.operation != gcp::thrift::Operation::Read)
      return false;
  if (sequence.__isset.sensor && sequence.sensor.__isset.deviceload){
  gcp::thrift::deviceloadRequest reqField = sequence.sensor.deviceload;

  if (!reqField.__isset.memory_load &&
      !reqField.__isset.cpu_load)
        reqField.__isset.memory_load = reqField.__isset.cpu_load = true;


  deviceload.__set_memory_load(sequence.memory_load);
  deviceload.__set_cpu_load(sequence.cpu_load);

  timers.__set_uptime(sequence.uptime);
  timers.__set_boot_time(sequence.boot_time);

  sensor.__set_deviceload(deviceload);
  sensor.__set_temperature(sequence.temperature);
  sensor.__set_timers(timers);
  responseSequence.__set_sensor(sensor);

  gcp_thrift_utils::fillResponseSequence(responseSequence,
                                         true,
                                         sequence.sequenceNumber,
                                         gcp::thrift::Operation::ReadResponse,
                                         gcp::thrift::ResponseCode::Ok);

  return true;
}
```

Рисунок 5.9 – Функція обробки запиту на оновлення метрики.

На рисунку 5.9 зображена функція, що обробляє запит на метрику sensors. Також функція заповнює thrift поля перед надсиланням запиту, та передає її в наступну функцію, котра повинна налагодити передачу данних між вузлами чи процесами.

Міжпроцесорна комунікація була реалізована на основі сокетів. Розгялнутий метод комунікації між девайсами та хостами також підходить для комунікації між процесами, тому даний підхід був використаний при реалізації модульної архітектури.

5.6 Розробка користувацького інтерфейсу

Для повного тестування функціоналу системи було створено демо додаток для телефону.

Було розроблено мінімальний користувацький інтерфейс. На рисунку 5.10 зображено екран логіну в додаток., для того , щоб пройти авторизацію потрібно мати власний аккаунт, який може бути створений тільки адміністратором.

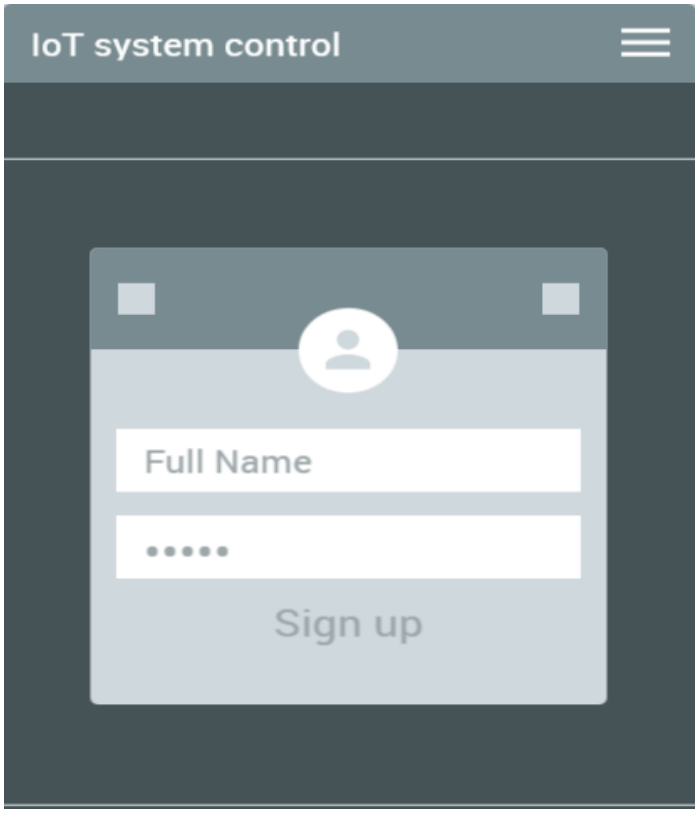

Рисунок 5.10 – Екран входу в додаток.

Після того як користувач успішно залогіниться в систему, його перенесе на екран з головним меню та короткою інформацією про профіль користувача. На рисунку 5.11 показано користувацький інтерфейс головного профілю користувача.

З самого верху екрану можна побачити головне меню, на якому розташовані такі елементи як metrics та settings.

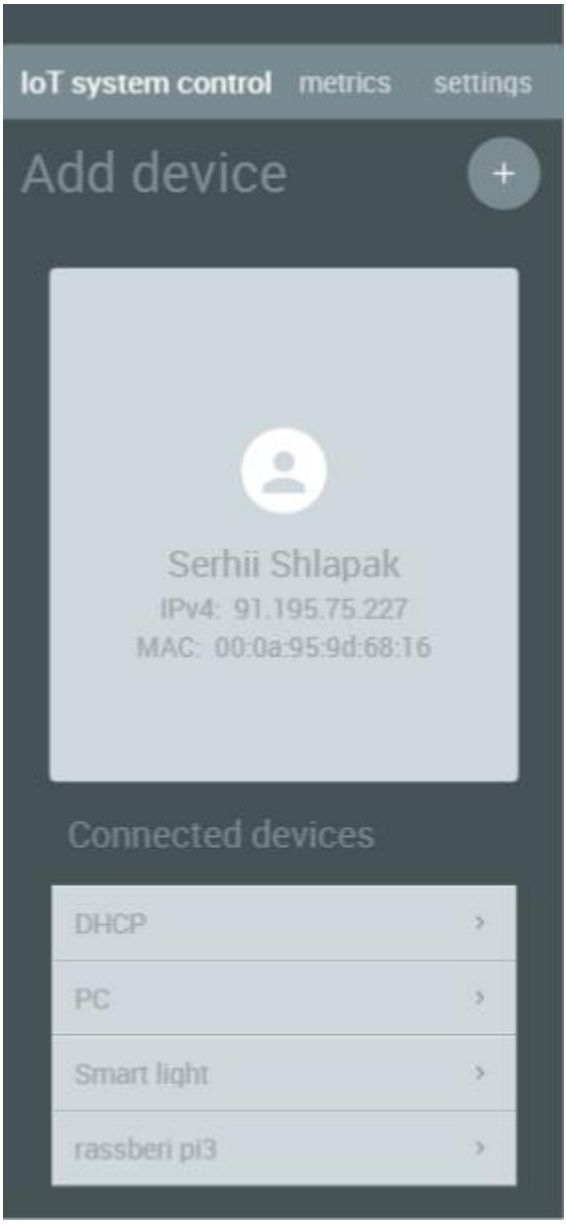

Рисунок 5.11 – Екран головного меню.

Також користувач перебуваючи в головному меню має можливість подивитись які саме девайси підключені до мережі. Двічі натиснувши на потрібний елемент

користувач потрапляє на екран з детальною інформацією про девайс. На рисунку 5.12 зображена детальна інформація про обраний девайс.

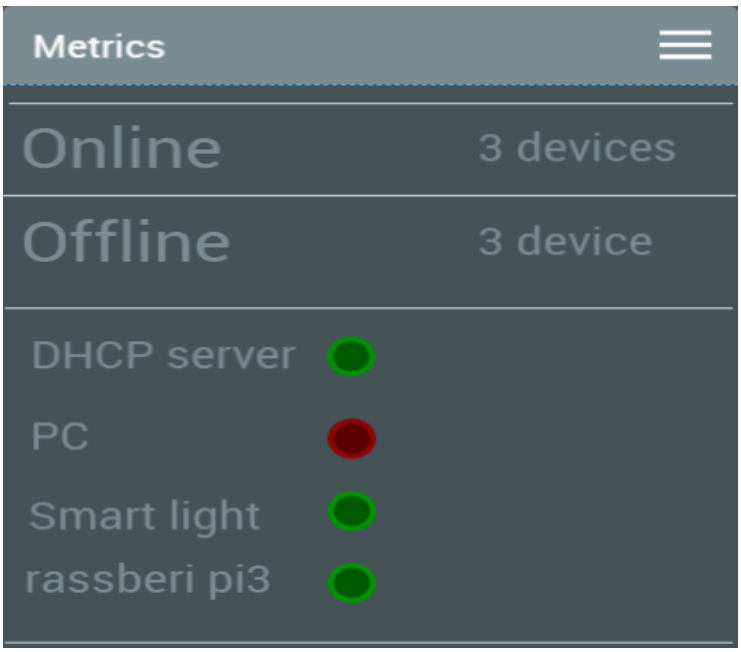

Рисунок 5.12 – Екран з детальна інформацією про девайс.

Для того, щоб перейти в розділ моніторингу та збору метрик, потрібно перейти з головного екрану вибравши потрібний елемент з головного меню зверху. На рисунку 5.13 показано екран моніторигу системи.

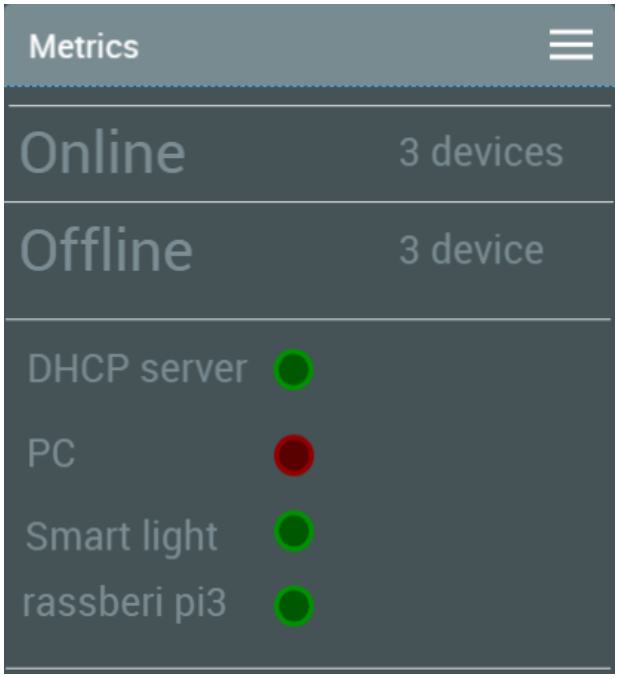

Рисунок 5.13 – Екран моніторингу системи.

Якщо користувачу портрібно переглянути детальну інформацію про стан девайсу в реальному часі, а також мати змогу дивитись за змінами показників в різні моменти часу, потрібно двічі клашнути на потрібний девайс з списку елементів системи. На рисунку 5.14 зображена детальна інформація про стан девайсу в даний момент часу.

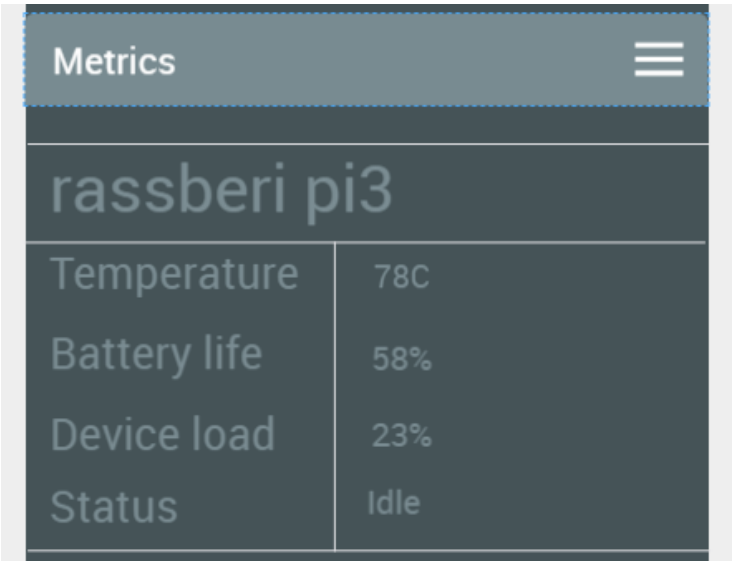

Рисунок 5.14 – Екран аналізу метрик.

Для того, щоб користувач зміг внести навіть мінімальні зміни в налаштування та конфігурацію системи, адміністратор повинен надати права редагування, дана операція буде розглянута в розділі нище. Зараз розглянемо ситуація коли у користувача є права для редагування, у цьому випадку користувач зможе змінювати налаштування системи.

На рисунку 5.15 видно список налаштувань які користувач з правами редагування може змніювати через користувацький додаток. Для більш радикальних змін, потрібно залогінитись в CLI адміністратору.

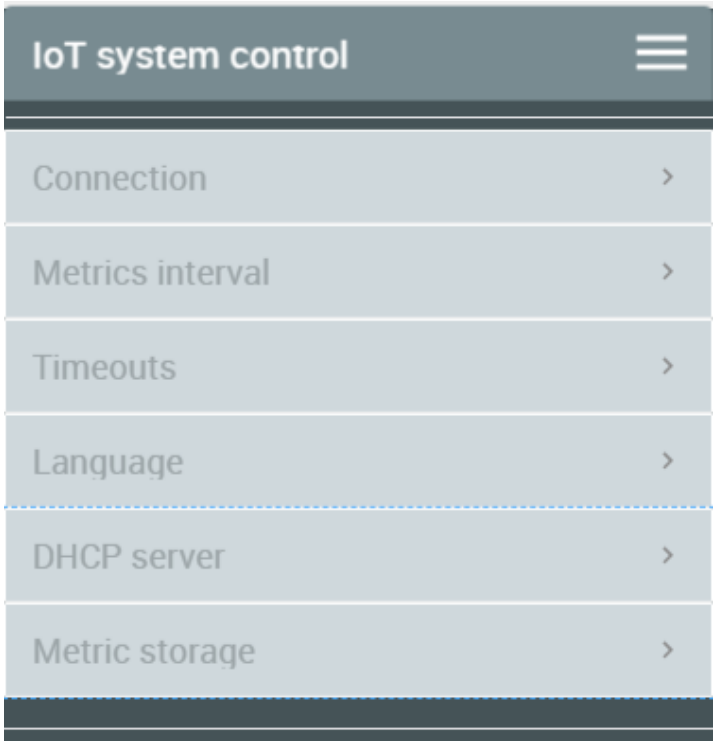

Рисунок 5.15 – Результат виконання команди перегляду стану девайсів.

5.7 Розробка користувацького інтерфейсу

Для того, щоб адміністратор зміг в повній мірі змінювати конфігурацію та налаштування, для покращення топології мережі, був розроблений CLI адмінінстратору. Він потрібний для того, щоб відокремити можливості мережевого адміністратору, який є спеціалістом, від рядового користувача. На рисунку 5.16 видно результат виконання команди перегляду статусу девайсів.

```
admin@SmartIoTSystem> show device status

VC:VS   MAC-ADDRESS    IP-ADDRESS                          STATUS          CLOCK
------- -------------- ------------------------------  ---------------    --------
1:0     0020.a324.1d03 200.200.61.84                        online          locked
2:0     0020.a324.92b2 200.200.61.83                        online          locked
3:0     0020.a325.f4ab 3000:200:200:61::a6                  online          locked
4:0     0020.a327.b1eb 3000:200:200:61::b2                  offline         unlocked
```

Рисунок 5.16 – Результат виконання команди перегляду стану девайсів.

Як видно з рисунку, адміністратор отримує інформацію про ip та mac адреси девайсів, що підключені до одінєї мережі, а також інформацію про їх статус, онлайн або оффлайн, та стан синхронізації з системою, поле CLOCK показує на те, зміг девайс синхронізуватись з системою використовуючи PTP протокол чи ні.

```
admin@SmartIoTSystem> configure secure stanadrt

1) IPsec
2) MACsec
3) Dot1x (802.1x)

dmin@SmartIoTSystem> configure secure stanadrt 3

Dot1x (802.1x) become main secure standart.
done...
```

Рисунок 5.17 – Результат виконання команди зміни стандарту захисту.

На рисунку 5.17 зображено результат виконання команди вибору стандарту захисту за замовчуванням, як бачимо для того, щоб команда правильно відпрацювала потрібно вибрати один з трьох доступних варіантів.

```
admin@SmartIoTSystem> reconfigure force

Lease files were changed.
System is going to set factory settings
Dot1x (802.1x) become main secure standart.
Wi-Fi become main transport tecnology
done...
```

Рисунок 5.18 – Результат виконання команди зміни стандарту захисту.

На рисунку 5.18 зображено результат виконання команди автоматичної переконфігурації системи.

6. РОЗРОБЛЕННЯ СТАРТАП ПРОЕКТУ

6.1 Опис ідеї проекту

Ідея проекту даної роботи - розробити систему, який облегшує конфігурацію, моніторинг та керування мережею інтернет речей. Це є надзвичайно позитивним фактором так як у сфера інтернет речей з кожним роком все більше і більше розвивається і маштабується.

Користувачі системи мають можливість створити мережі та підмережі інтернет речей, конфігурувати систему, моніторити зміни в реальному часі.

Узагальнення цих ідей можна побачити в таблиці 6.1.

Таблиця 6.1 Опис ідеї стартап проекту.

Зміст ідеї	Напрямки застосування	Вигоди для користувача
		Спрощення створення топології мережі для інтернет речей
		Можливість моніторингу та аналізу метрик мережі
		Покращений захист мережі та приватної інформації користувача

Основні вимоги до стартап-проектів, як правило, полягають у наступному:

- аналіз ринку на наявність аналогічних рішень;
- ознайомлення можливих споживачів та партнерів з стартап-проектом;
- вибір найкращого моменту для подання інформації про стартап-проект;
- забезпечення процесу обробки інформації при реалізації проекту;
- зберігання результату обробки інформації в необхідному вигляді.

Загальні обмеження на роботу з інформацією можуть бути представлені як група характеристик, які дійсно належать сторонам стартап-проекту, а саме:

- об'єкту (джерелу інформації);
- медіатору (засобу отримання, обробки, виробництва і передачі інформації, посереднику між суб'єктом і об'єктом);
- суб'єкту (активній стороні діяльності, одержувачу і перетворювачу інформації).

Зараз на ринку існує декілька основних сисктем керування мережами інтернет речей, доведеться конкурувати і з ними, а не тільки з тими, що пропонують рекомендації. Тому із найкрупніших конкурентів можна виокремити Google Cloud IoT, Asure Stream та AWS IoT Platform.

6.2 Аналіз потенційних техніко-економічних переваг

Аналіз сильних, слабких та нейтральних сторін стартап-проекту визначено в таблиці 6.2. Перевагою системи є покращений захист системи та користувацької інформації.

Таблиця 6.2 Визначення сильних, слабких та нейтральних характеристик ідеї проекту

Технічні та економічні характеристики	товари/концепції конкурентів			Слабка сторона	Нейтральна сторона	Сильна сторона
	Мій проект	Google Cloud IoT	Asure Stream			

Продовження таблиці 6.2

Можливість використовувати 802.1x захист	Так	Так	Так			+
Висока вартість	Ні	Так	Так	+		
Кількість підтримуваних вендорів	Мала	Середня	Середня			+
Можливість використання стандарту захисту macsec	Так	Ні	Ні			+
Можливість вибору комунікаційного протоколу	Так	Так	Ні			+
Можливість збору та аналізу метрик	Так	Так	Так			+
Час розгортування системи	35	32	36			+

Маштабування системи	+	+	+		+	

6.3 Технологічний аудит ідеї проекту

Список технологій реалізації програмного комплексу наведено в таблиці 6.3.

Таблиця 6.3 - Технологічна здійсненність ідеї проекту

№	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Реалізація прогрмамного забезпечення з використанням мови програмування C++.	Мова програмування C++, стандартна бібліотека STL, Boost.asio	Наявні	Доступні
2	Реалізація скриптів для автоматичної конфігурації системи з використанням мови програмування Python.	Мова програмування Python та Bash, утиліта cron	Наявні	Доступні
3	Реалізація міжпроцесорної	Фреймворк Apache Thrift,	Наявні	Доступні

	взаємодії з використанням фреймфорку Apache Thrift.	мова прогрмування Python та C++		

Продовження таблиці 6.3

№	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
4	Реалізація механізму збору та збереження метрик за допомогою мов програмування C++ та SQL.	Мова прогрмування C++ та SQL	Наявні	Доступні
5	Створення додатку з використанням мови програмування Python та фреймворку Kivy.	Мова прогрмування Python та фреймворк Kivy	Наявні	Доступні
6	Створення інтерфейсу адміністратору у вигляді командної строки з використанням мов	Мова прогрмування Python та C++, стандарта бібліотка STL	Наявні	Доступні

	програмування Python та C++.			

6.4 Аналіз ринкових можливостей запуску стартап-проекту

Огляд характеристик потенціального ринку систем керування мережами інтернет речей наведено в таблиці 6.4. Також проведено аналіз рентабельності галузі.

Таблиця 6.4 - Попередня характеристика потенційного ринку проекту

Показники стану ринку (найменування)	Характеристика
Кількість головних гравців, од	5
Загальний обсяг продаж, грн/ум.од	400 грн/ум.од
Динаміка ринку (якісна оцінка)	Зростає
Наявність обмежень для входу (вказати характер обмежень)	Недискримінаційні якісні
Специфічні вимоги до стандартизації та сертифікації	Відсутні
Середня норма рентабельності в галузі (або по	68,5%

ринку), %	

6.5 Визначення потенційних груп клієнтів

Цільова аудиторія даного стартап-проекту наведена у таблиці 6.5. Були охарактеризовані відмінності потенційних груп, а також вимоги до споживачів товару. Основна потреба, що формує ринок – бажання автоматизувати більшість процесів людської життєдіяльності, а все це призводить до маштабування мереж. Також тема захисту мережі стає все більш актуальнішою з кожним роком.

Таблиця 6.5 - Характеристика потенційних клієнтів стартап-проекту

Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
Бажання зменшення часу роботи програміста та приріст швидкості реалізації додатків	1. Малий середній та великий бізнес галузі IT	1. Ведення бізнесу на платформах інтернет-ресурсів	Можливість моніторингу та аналізу системи в реальному часі, надійний захист мережі та інформації користувача від стороніх лиць

6.6 Аналіз ринкового середовища

У таблиці 6.6 наведені головні фактори загроз та реакції, котрі допоможуть справитись з виникнувшими проблемами. У таблиці 6.7 наведені основні фактори можливостей та реакція компанії, що дозволить реалізувати дані можливості.

Значиний розвиток ринку інтернету речей за останні десять років говорить тільки про те, що системи управління IoT мережами, як ніколи актуальні, але це не відміняє такого фактору, як висока конкурентність, саме тому потрібно подумати також про можливі загрози та шляхи їх вирішення.

Таблиця 6.6 - Фактори загроз

Фактор	Зміст загрози	Можлива реакція компанії
Крадіжка інтелектуальної власності	Крадіжка ідеї або ключової інтелектуальної інновації	Відсудження прав інтелектуальної власності Забезпечення якісного захисту інформації Зміна методики групування даних Розробленя нових підходів реалізації

		продукту та в подальшому патентування
Відмова користувачів від використання продукту	Користувач відмовляється купляти продукт та технічну підтримку	Пропозиція більш вигідних умов та перегляд реалізації свого рішення
Недостатній стартовий капітал	Недостаній стартовий капітал для реалізації демо версії продукту	Пошук нових інвесторів

Таблиця 6.7 - Фактори можливостей

Фактор	Зміст можливості	Можлива реакція компанії
Отримання необхідних інвестицій	Сформований початковий капітал, необхідний для реалізації	Розробка демо версії продукту

	мінімально життєздатного продукту	
Висока зацікавленість користувачів	Обіг використання Залежить від кількості девайсів, що підключені до однієї мережу.	Підтримка стабільної роботи системи та розроблення нових інструментів, що покращать роботу системи та облегшать її використання
Успішна маркетингова політика	Результатом проведеної маркетингової стратегії є велика кількість нових користувачі та швидкий ріст зацікавленості в продукті	Підтримка стабільної роботи системи та проведення масштабування системи Збільшення цін на використання системи Використання подібної маркетингової стратегії надалі для залучення нових користувачів

Продовження таблиці 6.7

Ліквідація конкурента	Конкурент ліквідував свою компанію у	Проведення маркетингової кампанії для монополізації

	результаті власного бажання або зовнішніх чинників	ринку

6.7 Аналіз пропозиції

У таблиці 6.8 наведено ступеневий аналіз конкуренції на ринку (характеристики конкурентного середовища та його особоливості).

Таблиця 6.8 - Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Особливості конкурентного середовища В чому проявляється дана характеристика Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
Тип конкуренції - чиста	Немає чітко відокремленого лідера серед конкурентів, та відповідно мала їх кількість	Сприятливо впливає на розвиток проекту

Продовження таблиці 6.8

Рівень конкурентної боротьби -	Сфера зачіпає	Можливий швидкий вихід на світові ринки

міжнародний	практично всі країни світу та безліч галузей	
За галузевою ознакою - внутрішньогалузева	Загроза появи нових конкурентів Ринкова влада споживачів Висока потреба у товарі	Інформування ринку щодо якості використовуваної новаторської технології Пропозиція гнучких цін
За видами товарів - товарно-родові	Надання різних сервісів одного виду	Маркетингова політика
Цінова	Використання цін для покращення економічних умов збуту	Зменшення вартості сервісу Використання нових каналів розподілу

Аналіз конкуренції за М. Портером побудував більше детальну картину про стан ринку і можливості конкурентів (таблиця 6.9).

Таблиця 6.9 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари замінники

	Google Cloud IoT, Asure Stream та AWS IoT Platform	Цінність ідеї та її якісна реалізація	Немає	Споживачі прямо впливають на успішність продукту, оскільки сплачуються я кошти за ліцензію користуван ня	Конкуренти можуть стати дешевшими або стануть надавати якісніші послуги
Ви-сновки	Конкуренті в небагато, відсутність монополії, проте конкурент на боротьба значна за рахунок хорошої якості проектів конкуренті в	Є можливість виходу на ринок	Немає постачальник ів	Так, реалізація потреб клієнтів вирішує успішність проекту	Достатньо важко предентувати на клієнтів, які вже використовую ть інший готовий продукт

6.8 Перелік факторів конкурентоспроможності

Навіть при врахуванні конкурентів розроблена система має низку переваг, які наведені в таблиці 6.10.

Таблиця 6.10 — Обґрунтування факторів конкурентоспроможності

Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)

У конкурентів наявні функціональні проблеми реалізації	Поганий захист інформації про мережу та приватної інформації користувача
Ідея	Керування IOT мережею
Швидкий вихід на ринок	Достатньо випустити мінімальну базову працездатну версію продукту на ринок для його використання
Цінова політика	Отримання прибутку здійснюється за рахунок гнучкої моделі оплати

6.9 Аналіз сильних та слабких сторін стартап-проекту

За наведеними факторами конкурентоспроможності в таблиці 6.10 було виконано порівняльний аналіз сильних та слабких сторін програмного рішення (стартап-проекту), який відображено в таблиці 6.11.

Таблиця 6.11 — Порівняльний аналіз сильних та слабких сторін системи

Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні із системою управління процесами						
		-3	-2	-1	0	+1	+2	+3

У конкурентів наявні функціональні проблеми реалізації	18							+
Ідея	12					+		
Швидкий вихід на ринок	16					+		
Цінова політика	13					+		

6.10 SWOT-аналіз

Складений SWOT-аналіз на основі вище отриманих даних відображено в таблиці 6.12.

Таблиця 6.12 — SWOT-аналіз стартап-проекту

Сильні сторони: – централізоване управління великою кількістю девайсів – можливість моніторингу та аналізу мережі – широкий спектр функціоналу; – мала залежність від інвестицій; – невелика кількість сильних конкурентів;	Слабкі сторони: – потреба в рекламі – потреба спеціалізованих комплектуючих
Можливості: – невелика кількість конкурентів; – інвестиції;	Загрози: – вихід на ринок нових конкуренті – зміна тенденцій розвитку ринку

6.11 Опис цільових груп потенційних клієнтів

В таблиці 6.13 наведено характеристики потенційних клієнтів.

Таблиця 6.13 - Характеристика потенційних клієнтів стартап проекту

Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
IT команди	Висока	85%	Середня	Просто
Власне використання	Висока	90 %	Середня	Складно

6.12 Формування базових стратегій розвитку

В таблиці 6.14 наведено характеристики та визначення базової стратегії розвитку .

Базовою стратегією для стартап-проекту є стратегія диференціації, що означає надання програмному комплексу властивостей, які будуть відрізняють його від конкурентів.

Таблиця 6.14 — Визначення базової стратегії розвитку

Обрана альтернатива	Стратегія охоплення ринку	Ключові конку-рентоспроможні	Базова стратегія

розвитку проекту		позиції	розвитку
Пропонування програмного комплексу для IT команд	Диференційований маркетинг	Здатність пропонувати унікальний функціонал	Стратегія диференціації

6.13 Вибір стратегії конкурентної поведінки

В таблиці 6.15 наведено базові стратегії конкурентної поведінки.

Таблиця 6.15 — Визначення базової стратегії конкурентної поведінки

Чи є проект першопрохідцем на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
Ні	Залучати нових та забирати існуючих	Метрики та Користувацький інтерфейс для легшої інтеграції з іншими вендорами	Стратегія велику лідера

За допомогою зібраної інформації про потреби користувачу в даному сегменті ринку, а також після аналізу слабких та сильних сторін конкурентів в таблиці 6.16 наведено визначені стратегії позиціонування.

Таблиця 6.16 — Визначення стратегії позиціонування

Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного
Швидкодія, захищеність та зручність керування, налаштування системи	Стратегія диференціації	Розширення базового функціоналу	Якісний продукт, зрозумілий у використанні

Першим кроком до формування маркетингової концепції товару, який отримує споживач.

4.6.1 Формування маркетингової концепції

В таблиці 6.17 визначено ключові переваги концепції потенційного товару

Таблиця 6.17 – Визначення ключових переваг концепції потенційного товару

Потреба	Вигода, яку пропонує	Ключові переваги перед

	товар	конкурентами (існуючі або такі, що потрібно створити
Захищеність	Надійний захист від втручання зовні	У існуючих систем присутні проблеми захисту інформації, так як здебільшого вони використовують тільки один з стандартів захисту
Моніторинг та управління	Збір метрик з девайсів підключених до мережі та можливість управління ними	Більшість з існуючих систем хоч і мають реалізований механізм збору інформації та управління системою, але часто він занадто важкий у щоденному використанні

Наступним кроком є визначення цінових меж представлених в таблиці 6.18, якими необхідно керуватись при встановленні ціни на товар

Таблиця 6.18 – Визначення меж встановлення ціни

Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
Товари замінники мають високий рівень цін	Товарів аналогів немає	Цільова група споживачів має високий рівень доходів	Нижня межа – це найменша ціна рішення найближчого найдешевшого конкурента, верхня межа, це на 1%

			більша ціна на найближчого найдорожчого конкурента

ВИСНОВКИ

Розвиток технології інтернет речей стрімкий та неменучий, технолгія розповсюджуєтсья по всім галузям людсьокої життєдіяльності, починаючи від автоматизації власного будинку, так званий розумний дім, та закінчуючи автоматизацією індустріальних галузей. В даній роботі було вирішено одну з найголовніших проблем, яка присутня у всіх системах керування мережами інтернету речей, а саме проблема захищеності мережі та приватної інформації користувача від посторонніх лиць.

Реалізована система управління мережею інтернет речей є обєднанням технологій та програмних засобів, які позитивно сприяють на підвищення захисту інформації та мережі в цілому. Також було розроблено можливість збору та аналізу метрик та можливість вибору протоколу комунікації.

Метою дослідження було покращення захищеності мережі та інформації користувача, для досягнення мети були вирішені наступні задачі:

- досліджено особливості стандартів захисту мереж інтернету речей
- проведено аналіз існуючих рінешь та методів захисту які вони використовують
- обрано технологічний стек для вирішення задач
- спроектовано базу даних та архітектуру системи управління IoT мережею
- реалізовано механіз збору метрик та їх аналізу
- побудовано інтерфейс користувача та адміністратора

Висновки, що були зроблені при виконанні магістрської дисертації:

- система повинна складатись з декількох основних підсистема, а саме: користувацького додатку, CLI адміністратора та головного серверу з усією бізнес логікою та реляційною базою даних
- для покращення захисту мережі необхідно використати наступні стандарти захисту:
  - IPsec
  - MACsec
  - 802.1x
- для оптимізації комунікації між процесами потрібно використовувати фреймворк Apache Thrift
- для практичного використання системи потрібно розробити детальний бізнес план та запровадити відповідний стартап процес.

Кінцевим результатом розробки системи управління мережею інтеренту речей повинна стати реалізація стартап-проекту на основі отриманих результатів, а також впровадження виготовленого рішення на ринок систем управління.

Опис стартап-проекту допоміг більш детальніше та ширше оглянути розроблене рішення, переваги які повинна зберегти система, а також недоліки, які повинні бути вирішені перед запуском продукту. Вся ця інформація була закріплена у відповідних таблицях розділу розроблення стартап проекту.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1) Корнієнко Б.Я. Безпека інформаційно-комунікаційних систем та мереж/ Б.Я. Корнієнко // Навчальний посібник для студентів спеціальності 125 «Кібербезпека». – К.НАУ, 2018. – 226 с.
2) IoT future [Електронний ресурс]: Режим доступу: https://www.ericsson.com/en/future-technologies/future-iot
3) The IOT PIE [Електронний ресурс]: Режим доступу: https://blog.enocean.com/the-iot-pie-how-to-grab-your-slice/
4) Trends to Watch in Cloud Computing for 2017 [Електронний ресурс]: Режим доступу: https://www.billchamberlin.com/horizonwatching-blog-2/
5) IoT World Today [Електронний ресурс]: Режим доступу: https://www.iotworldtoday.com/subject/energy/
6) 20 Surprising IoT Statistics You Don’t Already Know [Електронний ресурс]: Режим доступу: https://www.thesslstore.com/blog/20-surprising-iot-statistics-you-dont-already-know/
7) Корниенко Б.Я. Информационная безопасность и технологии компьютерных сетей: монография / Б.Я. Корниенко // ISBN 978-3-330-02028-3, LAMBERT Academic Publishing, Saarbrucken, Deutschland. − 2016. – 102 с.
8) Azure Stream Analytics [Електронний ресурс]: Режим доступу: https://azure.microsoft.com/ru-ru/services/stream-analytics/
9) Google Cloud IoT [Електронний ресурс]: Режим доступу: https://cloud.google.com/solutions/iot/
10) AWS IoT SDKs [Електронний ресурс]: Режим доступу: https://docs.aws.amazon.com/iot/latest/developerguide/iot-sdks.html
11) ThingWorx Industrial IoT Platform [Електронний ресурс]: Режим доступу: https://developer.thingworx.com/platform
12) Bosch IoT Gateway [Електронний ресурс]: Режим доступу: https://developer.bosch-iot-suite.com/service/gateway-software/

13) Корниенко Б.Я. Кибернетическая безопасность – операционные системы и протоколы / Б.Я. Корниенко // ISBN 978-3-330-08397-4, LAMBERT Academic Publishing, Saarbrucken, Deutschland. − 2017. – 122 с.

14) Корнієнко Б.Я. Дослідження моделі взаємодії відкритих систем з погляду інформаційної безпеки /Б.Я. Корнієнко//Наукоємні технології. – 2012, № 3 (15), С. 83 – 89, doi.org/10.18372/2310-5461.15.5120 (ukr).

15) Korniyenko B.Y. Open systems interconnection model investigation from the viewpoint of information security /B. Korniyenko, O. Yudin, E. Novizkij//The Advanced Science Journal. – 2013. - issue 8. - P. 53 – 56.

16) Bluetooth Broadcasting [Електронний ресурс]: Режим доступу: https://www.c-sharpcorner.com/UploadFile/2cb323/bluetooth-broadcasting/

17) Корнієнко Б.Я. Реалізація інформаційної безпеки у моделі взаємодії відкритих систем/Юдін О.К., Корнієнко Б.Я./Збірник тез VI Міжнародної науково-технічної конференції «Комп'ютерні системи та мережні технології» (CSNT-2013), 11-13 червня 2013 р., - С.73.

18) ZigBee [Електронний ресурс]: Режим доступу: http://www.rovdo.com/zigbee

19) Корнієнко Б.Я. Інформаційні технології оптимального управління виробництвом мінеральних добрив : монографія / Б.Я. Корнієнко. – К.: Вид-во Аграр Медіа Груп, 2014. – 288 с.

20) Korniyenko B.Y. Design and research of mathematical model for information security system in computer network / B.Y. Korniyenko, L.P. Galata // Науковий журнал «Наукоємні технології». – 2017, № 2 (34), С. 114 - 118.

21) Korniyenko B. Modeling of security and risk assessment in information and communication system /B. Korniyenko, L. Galata, O. Kozuberda/ Sciences of Europe. – 2016. – V. 2. – No 2 (2). – P. 61 -63.

22) Korniyenko B. The classification of information technologies and control systems / B. Korniyenko // International scientific journal. – 2016. –№ 2. – P. 78 - 81.

23) Korniyenko B. Risk estimation of information system / B. Korniyenko, A. Yudin, L. Galata// Wschodnioeuropejskie Czasopismo Naukowe. – 2016. –№ 5. – P. 35 - 40.

24) Корнієнко Б.Я. Безпека аутентифікації у web-ресурсах / Б.Я. Корнієнко, О.К. Юдін, О.С. Снігур / Науково-практичний журнал «Захист інформації». – 2012. - № 1 (54). - С. 20 -25, doi.org/10.18372/2410-7840.14.2056 (ukr).

25) PostgreSQL vs. MySQL [Електронний ресурс]: Режим доступу: http://www.postgresqltutorial.com/postgresql-vs-mysql/

26) Redis vs. MongoDB [Електронний ресурс]: Режим доступу: https://scalegrid.io/blog/comparing-in-memory-databases-redis-vs-mongodb-percona-memory-engine/

27) Корнієнко Б.Я. Прикладні програми управління інформаційними ризиками / Б.Я. Корнієнко, Ю.О. Максімов, Н.М. Марутовська / Науково-практичний журнал «Захист інформації». – 2012. - № 4 (57). - С. 60 – 64, doi.org/10.18372/2410-7840.14.3493 (ukr).

28) Korniyenko B. Security Estimation of the Simulation Polygon for the Protection of Critical Information Resources / B. Korniyenko, L. Galata, L. Ladieva //CEUR Workshop Proceedings, Selected Papers of the XVIII International Scientific and Practical Conference "Information Technologies and Security" (ITS 2018) Kyiv, Ukraine, November 27, 2018, Vol-2318, - P. 176-187, urn:nbn:de:0074-2318-4

29) Корнієнко Б.Я. Дослідження імітаційного полігону захисту критичних інформаційних ресурсів методом IRISK / Б.Я. Корнієнко, Л.П. Галата // Моделювання та інформаційні технології. – 2018. – Вип. 83. – С. 34-41.

30) Корнієнко Б.Я. Побудова та тестування імітаційного полігону захисту критичних інформаційних ресурсів / Б.Я. Корнієнко, Л.П. Галата // Науковий журнал «Наукоємні технології». – 2017, № 4 (36), С. 316 - 322.

31) Корнієнко Б.Я. Інформаційні технології оптимального управління виробництвом мінеральних добрив :монографія / Б.Я. Корнієнко. – К.: Вид-во Аграр Медіа Груп, 2014. – 288 с.

ДОДАТКИ

Додаток А – Публікація

www.konferenciaonline.org.ua

Міжнародна наукова інтернет-конференція

"Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення" (випуск 43)

14 листопада 2019 р.

Частина 1

Частина 1

Секція 1. Інформаційні системи і технології

Шлапак С.С.
Порівняння стандартів безпеки комп'ютерних мереж.……...…123

Шматок В.В., Сватюк О.Я.
Технологія доповненої реальності.……...…124

Юскович-Жуковська В.І., Соловей Л.Я.
Напрямки використання соціальних мереж у підготовці ІТ-фахівців……...126

Mykytas A.O., Kononenko O.M., Bratyshchenko M.R.
Інформаційна безпека.……...…128

Mykytas A.O., Kononenko O.M., Bratyshchenko M.R.
Соціальна інформатика.……..…130

Mykytas A.O., Kononenko O.M., Bratyshchenko M.R.
Технології інтернету речей (IoT).……....…131

Шлапак С.С.
Національний Технічний Університет України "Київський політехнічний інститут ім. Ігоря Сікорського", м. Київ
Кафедра автоматики та управління в технічних системах, студент

ПОРІВНЯННЯ СТАНДАРТІВ БЕЗПЕКИ КОМП'ЮТЕРНИХ МЕРЕЖ

У захищеній від MACsec мережі кожен вузол має щонайменше один захищений канал передачі. Цей захищений канал передачі пов'язаний з ідентифікатором: ідентифікатором захищеного каналу (SCI). Захищений канал передачі також зберігає різні параметри конфігурації, наприклад, чи потрібно виконувати захист відтворення або включити шифрування. Кожен вузол, який очікує отримання трафіку, що надсилається через певний захищений канал передачі, повинен налаштувати відповідний захищений канал прийому. Цей прийом захищеного каналу повинен мати SCI, відповідний SCI захищеного каналу передачі однорангового.

У межах кожного захищеного каналу (і передачі, і прийому) визначаються захищені асоціації. Захищені асоціації містять ключі шифрування та ідентифікуються за номером їх асоціації. З кожним захищеним об'єднанням пов'язаний ще один важливий параметр: номер пакету. На стороні передачі цей номер пакета вводиться в заголовок MACsec і використовується в процесі шифрування. На стороні прийому номер пакета із заголовка MACsec може бути перевірений на номер пакету, локально збережений у відповідній захищеній асоціації, щоб виконати захист відтворення.

IPsec

Перш за все, MACsec і IPsec працюють на різних мережевих шарах. IPsec працює над IP-пакетами, на рівні 3, тоді як MACsec працює на рівні 2, на кадрах Ethernet. Таким чином, MACsec може захищати весь трафік DHCP та ARP, який IPsec не може захистити. З іншого боку, IPsec може працювати через маршрутизатори, тоді як MACsec обмежений локальною мережею.

Як MACsec, так і IPsec, користувацькі програми не потребують змін, щоб скористатися гарантіями безпеки, які надають ці стандарти.У Red Hat Enterprise Linux підтримка IPsec надається libreswan

SSL/TLS

SSL / TLS працює на ще одному шарі, а саме на п'ятому (додатковому) шарі. Таким чином, безпека програм може вимагати змін програми, що може виявитися досить проблематичним. З іншого боку, вставлення

криптографічного шару безпосередньо в додатки також пропонує деякі переваги. Програмне забезпечення може самостійно перевіряти стан шифрування та справжність на основі політики та може відповідно реагувати. Перевірка цих властивостей безпеки безпосередньо в додатку може забезпечити захист від кінця до кінця. У Red Hat Enterprise Linux SSL / TLS надається декількома бібліотеками, такими як GnuTLS, NSS, OpenJDK / JSSE та OpenSSL (в алфавітному порядку).

Список використаних джерел:

1.Enhancing Network Security with MACsec [Електронний ресурс] – Режим доступу: https://www.curtisswrightds.com/news/blog/enhancing-network-security-with-macsec.html
2.What is ipsec ? [Електронний ресурс] – Режим доступу: https://searchsecurity.techtarget.com/definition/IPsec-Internet-Protocol-Security
3.Transport Layer Security [Електронний ресурс] – Режим доступу: https://en.wikipedia.org/wiki/Transport_Layer_Security